UNIVERSITÉ DE MONTRÉAL

REYNOLDS-AVERAGED NAVIER-STOKES BASED ICE ACCRETION FOR AIRCRAFT WINGS

KAZEM HASANZADEH LASHKAJANI

DÉPARTEMENT DE GÉNIE MÉCANIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION

DU DIPLÔME DE PHILOSOPHIAE DOCTOR

(GÉNIE MÉCANIQUE)

DÉCEMBRE 2015

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

REYNOLDS-AVERAGED NAVIER-STOKES BASED ICE ACCRETION FOR AIRCRAFT WINGS

présentée par : HASANZADEH LASHKAJANI Kazem

en vue de l'obtention du diplôme de : Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. TRÉPANIER Jean-Yves, Ph.D., président

M. PARASCHIVOIU Ion, Doctorat, membre et directeur de recherche

M. LAURENDEAU Eric, Ph.D., membre et codirecteur de recherche

M. CAMARERO Ricardo, Ph.D., membre

M. MORENCY François, Ph.D., membre externe

DEDICATION

To my family and loved ones

ACKNOWLEDGEMENTS

The work has been performed through a Collaborative R&D Grant No. 341083–06 with Bombardier Aerospace and the Natural Sciences and Engineering Research Council of Canada (NSERC).

First, I would like to express my gratitude to my research directors, Professor Eric Laurendeau and Professor Ion Paraschivoiu for their support of my Ph. D. study.

Particularly, I would like to express my sincere appreciation to Professor Eric Laurendeau, for his patience, knowledge, motivation and guidance through the entire course of my research study. An outstanding teacher, who has taught me the insight in fluid mechanical engineering and its industrial applications.

I would like to thank Dr. Sina Arabi, my friend, who has helped me to understand the knowledge of mesh generation.

An appreciation to BBA, advance aero department, especially to Mr. Corentin Brette, Dr. Patrick Germain, Dr. Alberto Pueyo, Dr. Guy Fortin, Dr. Kurt Sermeus, Mrs. Sherry Vafa for their help and support.

And at last, I would like to thank my friends and colleagues for the happy moments we spent during the study at Polytechnique Montreal.

RÉSUMÉ

Cette thèse aborde l'un des enjeux actuels de la sécurité aérienne pour augmenter les capacités de simulation de givrage pour la prédiction des formes complexes 2D et 3D de verglas sur les surfaces des aéronefs. Durant les années 1980 et 1990, le domaine de l'aérogivrage numérique s'est développé pour soutenir la conception et la certification des aéronefs volant dans des conditions givrantes. Les technologies multidisciplinaires utilisées dans ces codes étaient : l'aérodynamique (méthode des panneaux), le calcul des trajectoires des gouttelettes (méthode Lagrangienne), le module thermodynamique (modèle Messinger) et le module de géométrie (accumulation de glace). Ceux-ci sont intégrés dans un module quasi-stationnaire pour simuler le processus d'accumulation de glace en fonction du temps (procédure à plusieurs pas de temps). Les objectifs de la présente recherche visent à améliorer le module aérodynamique en passant de Laplace à un solveur d'équations de Navier-Stokes moyennée (*RANS*). Les avantages sont nombreux. Tout d'abord, le modèle physique permet le calcul des effets visqueux dans le module aérodynamique. Deuxièmement, la solution du programme d'aérogivrage fournit directement les moyens pour caractériser les effets aérodynamiques du givrage, comme la perte de portance et la traînée accrue. Troisièmement, l'utilisation d'une approche de volumes finis pour résoudre les équations aux dérivées partielles (*PDE*) permet des analyses rigoureuses de convergence en maillage et en temps. Enfin, les approches développées en 2D peuvent être facilement transposées aux problèmes 3D.

La recherche a été réalisée en trois étapes principales, chacune fournissant des aperçus des approches numériques globales. La réalisation la plus importante vient de la nécessité de développer des algorithmes de génération de maillage spécifiquement pour assurer des solutions réalisables en plusieurs étapes de calculs très complexes d'aéro-givrage. Les contributions sont présentées dans l'ordre chronologique de leurs réalisations.

D'abord, un nouveau cadre de simulation de glace bidimensionnel basé sur un code RANS, CANICE2D-NS, est développé. Un code RANS à maillage à simple bloc de l'université de Liverpool (nommé SMB) fournit la solution aérodynamique en utilisant le modèle de turbulence Spalart-Allmaras. L'outil commercial ICEM CFD est utilisé pour le remaillage du profil glacé pour le lissage du domaine. Le nouveau couplage est entièrement automatisé et capable d'effectuer des simulations d'accumulation de la glace à plusieurs pas de temps par une

approche quasi-stationnaire. En outre, le logiciel permet une analyse de l'écoulement de l'air et la prévision des performances aérodynamiques des profils glacés. La convergence de l'algorithme quasi-stationnaire est vérifiée et identifie la nécessité de l'augmentation d'un ordre de grandeur dans le nombre de pas de temps dans les simulations de givrage afin de parvenir à des solutions indépendantes du nombre d'incréments de temps.

Deuxièmement, un code Navier-Stokes à maillages à blocs multiples, NSMB, est couplé avec le programme de givrage CANICE2D. L'attention est accordée à l'implémentation du modèle de rugosité ONERA dans le modèle de turbulence Spalart-Allmaras et à la convergence de la procédure itérative stationnaire et quasi-stationnaire. Les effets d'une rugosité de surface uniforme dans la simulation quasi-stationnaire de l'accumulation de glace sont analysés à travers différents cas de validation. Les résultats de CANICE2D-NS montrent un bon accord avec les données expérimentales à la fois en termes de formes de glace prédites ainsi qu'en termes d'analyses aérodynamiques des formes de glace prédites et expérimentales.

Troisièmement, un code Navier-Stokes structuré à maillage à simple bloc, NSCODE, est couplé avec le cadre de givrage CANICE2D-NS. L'attention est accordée à l'implémentation de la rugosité du modèle Boeing dans le modèle de turbulence Spalart-Allmaras, et à l'accélération de la convergence des procédures itératives stationnaires et quasi-stationnaires. Les effets d'une rugosité de surface uniforme dans la simulation de l'accumulation quasi-stationnaire de glace sont analysés à travers différents cas de validation et de comparaisons entre codes avec le même programme couplé avec le solveur Navier-Stokes NSMB. L'efficacité de l'approche à grilles multiples dans la direction J est démontrée pour la résolution des équations d'écoulement d'air sur des géométries glacées complexes.

Puisqu'il a été remarqué à travers les différents résultats obtenus que le logiciel de génération de maillage ICEM-CFD produit un certain nombre de problèmes tels que la mauvaise qualité des maillages et les carences en lissage (notamment les chocs de mailles), une quatrième étude propose un nouvel algorithme de génération de maillages. Un code de génération de maillages en plusieurs blocs et basé sur la résolution d'équations aux dérivées partielles, NSGRID, est développé à cet effet. L'étude comprend les développements de nouveaux algorithmes de génération de maillage sur des formes complexes de verglas contenant des géométries d'accumulation de glace à courbure hautement variable, comme des cornes de glace

simples ou doubles. Une approche en deux étapes consiste à discrétiser la surface de la géométrie, puis le domaine de résolution. Un algorithme curviligne adaptatif contrôlant la courbure est construit en résolvant une équation elliptique 1D avec termes sources périodiques. Cette méthode contrôle l'espacement de la longueur d'arc des mailles sur la surface de telle sorte que les régions de forte courbure convexe et concave autour des cornes de glace sont captées de manière appropriée. Il est montré que cette méthode traite efficacement le problème de choc de mailles. Ensuite, une nouvelle méthode mixte est développée en définissant des combinaisons de termes sources avec des équations elliptiques 2D. Les termes sources comprennent deux fonctions de contrôle communes, Sorenson et Spekreijse, et un troisième terme source supplémentaire pour forcer l'orthogonalité. Cette méthode mixte s'avère être très efficace pour améliorer la qualité des mailles d'un maillage complexe de verglas avec une résolution RANS. La performance en termes de réduction des résidus par itérations non linéaires de plusieurs algorithmes de résolution (Point-Jacobi, Gauss-Seidel, ADI, Point et Line SOR) est discutée dans le contexte d'un opérateur multi grilles complet. Des détails sont donnés sur les différentes formulations utilisées dans le procédé de linéarisation. Il est montré que la performance de l'algorithme de solution dépend de la nature de la fonction de contrôle utilisée. Enfin, les algorithmes sont validés sur des formes de glace expérimentales standards et complexes, démontrant l'applicabilité des méthodes.

Finalement, un programme bidimensionnel de simulation d'accumulation de glace basé sur la résolution RANS et automatisé pour le calcul par plusieurs incréments de temps, CANICE2D-NS, est développé et couplé avec un code CFD Navier-Stokes multi blocs, NSCODE2D, un code de génération elliptique de maillages multi blocs, NSGRID2D, et un solveur multi blocs Eulérien pour la trajectoire des gouttelettes, NSDROP2D (développé à l'École Polytechnique de Montréal). Le programme permet des calculs Lagrangiens et Eulériens de trajectoires de gouttelettes, la dernière profitant d'une approche de maillages superposés pour traiter les géométries à éléments multiples. Le code a été testé sur des cas de validation confidentiels et publics, y compris les cas standards NATO. En outre, une accélération d'un facteur allant jusqu'à 10 est observée dans la procédure de génération de maillages en utilisant un lisseur implicite avec une procédure à grilles multiples. Les résultats démontrent les avantages et la robustesse du nouveau programme dans la prédiction des formes de glace et des paramètres de performances aérodynamique.

ABSTRACT

This thesis addresses one of the current issues in flight safety towards increasing icing simulation capabilities for prediction of complex 2D and 3D glaze ice shapes over aircraft surfaces. During the 1980's and 1990's, the field of aero-icing was established to support design and certification of aircraft flying in icing conditions. The multidisciplinary technologies used in such codes were: aerodynamics (panel method), droplet trajectory calculations (Lagrangian framework), thermodynamic module (Messinger model) and geometry module (ice accretion). These are embedded in a quasi-steady module to simulate the time-dependent ice accretion process (multi-step procedure). The objectives of the present research are to upgrade the aerodynamic module from Laplace to Reynolds-Average Navier-Stokes equations solver. The advantages are many. First, the physical model allows accounting for viscous effects in the aerodynamic module. Second, the solution of the aero-icing module directly provides the means for characterizing the aerodynamic effects of icing, such as loss of lift and increased drag. Third, the use of a finite volume approach to solving the Partial Differential Equations allows rigorous mesh and time convergence analysis. Finally, the approaches developed in 2D can be easily transposed to 3D problems.

The research was performed in three major steps, each providing insights into the overall numerical approaches. The most important realization comes from the need to develop specific mesh generation algorithms to ensure feasible solutions in very complex multi-step aero-icing calculations. The contributions are presented in chronological order of their realization.

First, a new framework for RANS based two-dimensional ice accretion code, CANICE2D-NS, is developed. A multi-block RANS code from U. of Liverpool (named PMB) is providing the aerodynamic field using the Spalart-Allmaras turbulence model. The ICEM-CFD commercial tool is used for the iced airfoil remeshing and field smoothing. The new coupling is fully automated and capable of multi-step ice accretion simulations via a quasi-steady approach. In addition, the framework allows for flow analysis and aerodynamic performance prediction of the iced airfoils. The convergence of the quasi-steady algorithm is verified and identifies the need for an order of magnitude increase in the number of multi-time steps in icing simulations to achieve solver independent solutions.

Second, a Multi-Block Navier-Stokes code, NSMB, is coupled with the CANICE2D icing framework. Attention is paid to the roughness implementation of the ONERA roughness model within the Spalart-Allmaras turbulence model, and to the convergence of the steady and quasi-steady iterative procedure. Effects of uniform surface roughness in quasi-steady ice accretion simulation are analyzed through different validation test cases. The results of CANICE2D-NS show good agreement with experimental data both in terms of predicted ice shapes as well as aerodynamic analysis of predicted and experimental ice shapes.

Third, an efficient single-block structured Navier-Stokes CFD code, NSCODE, is coupled with the CANICE2D-NS icing framework. Attention is paid to the roughness implementation of the Boeing model within the Spalart-Allmaras turbulence model, and to acceleration of the convergence of the steady and quasi-steady iterative procedures. Effects of uniform surface roughness in quasi-steady ice accretion simulation are analyzed through different validation test cases, including code to code comparisons with the same framework coupled with the NSMB Navier-Stokes solver. The efficiency of the J-multigrid approach to solve the flow equations on complex iced geometries is demonstrated.

Since it was noted in all these calculations that the ICEM-CFD grid generation package produced a number of issues such as inefficient mesh quality and smoothing deficiencies (notably grid shocks), a fourth study proposes a new mesh generation algorithm. A PDE based multi-block structured grid generation code, NSGRID, is developed for this purpose. The study includes the developments of novel mesh generation algorithms over complex glaze ice shapes containing multi-curvature ice accretion geometries, such as single/double ice horns. The twofold approaches tackle surface geometry discretization as well as field mesh generation. An adaptive curvilinear curvature control algorithm is constructed solving a 1D elliptic PDE equation with periodic source terms. This method controls the arclength grid spacing so that high convex and concave curvature regions around ice horns are appropriately captured and is shown to effectively treat the grid shock problem. Then, a novel blended method is developed by defining combinations of source terms with 2D elliptic equations. The source terms include two common control functions, Sorenson and Spekreijse, and an additional third source term to improve orthogonality. This blended method is shown to be very effective for improving grid quality metrics for complex glaze ice meshes with RANS resolution. The performance in terms of residual reduction per non-linear iteration of several solution algorithms (Point-Jacobi, Gauss-

Seidel, ADI, Point and Line SOR) are discussed within the context of a full Multi-grid operator. Details are given on the various formulations used in the linearization process. It is shown that the performance of the solution algorithm depends on the type of control function used. Finally, the algorithms are validated on standard complex experimental ice shapes, demonstrating the applicability of the methods.

Finally, the automated framework of RANS based two-dimensional multi-step ice accretion, CANICE2D-NS is developed, coupled with a Multi-Block Navier-Stokes CFD code, NSCODE2D, a Multi-Block elliptic grid generation code, NSGRID2D, and a Multi-Block Eulerian droplet solver, NSDROP2D (developed at Polytechnique Montreal). The framework allows Lagrangian and Eulerian droplet computations within a chimera approach treating multi-elements geometries. The code was tested on public and confidential validation test cases including standard NATO cases. In addition, up to 10 times speedup is observed in the mesh generation procedure by using the implicit line SOR and ADI smoothers within a multigrid procedure. The results demonstrate the benefits and robustness of the new framework in predicting ice shapes and aerodynamic performance parameters.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

C_L	Lift coefficient
C_D	Drag coefficient
C_f	Skin friction coefficient
C_p	Pressure coefficient
AOA	Angle of attack
LWC	Liquid water content (kg/m^3)
MED	Mean droplet diameter (m)
w	Wall value
$\tilde{\nu}$	Transported quantity in the S-A model
d	Distance to the nearest wall (m)
d_0	Offset in the wall distance to account for wall roughness (m)
k_s	Equivalent sand grain roughness height (m)
K_s	Equivalent sand grain roughness height normalized by chord
k	Von Karman constant
$\tilde{S}$	Transformed vorticity
y	Distance along the wall normal (m)
y^+	Normalized wall distance yu_τ/ν
u	Longitudinal velocity component (m/s)
ν_t	Turbulent eddy viscosity (m^2/s)
u_τ	Friction velocity (m/s)
$h_{c,t}$	Turbulent convective heat transfer coefficient (W/m^{2}K)
St	Stanton number

ρ	Density (kg/m^3)
u_e	Boundary-layer edge velocity (m/s)
St_k	Roughness Stanton number
ν	Kinematic viscosity (m^2/s)
Pr_t	Turbulent Prandtl number
dt or Δt	Implicit scheme time step
ω	Relaxation factor
S	Source term
r	Geometry points arc length
Δs	Grid minimum spacing close to wall
ξ, η	Non-dimensional computational space coordinate
s, t	Non-dimensional parametric space coordinate
x, y	Physical space coordinate
i, j	Space coordinate indexes
P, Q	Domain control functions
J	Jacobian
X	Point position vector
n	Unit surface normal
δ	Specified distance distribution
H_0, H_1	Cubic hermit interpolation functions
g_{11}, g_{12}, g_{22}	Covariant metric components (General Poisson equations)
α, β, γ	Covariant metric components (Sorenson equations)
a, b, c, d	Damping ratio factors

p, q, r, s	Surface control functions
a^{11}, a^{12}, a^{22}	Covariant metric components (Spekreijse equations)
p_{11}, p_{12}, p_{22}	Control functions parameters

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 Context

One of the current challenges towards increasing our knowledge of aircraft icing effects is the development of numerical algorithms for predicting complex 3D glaze ice shapes [1-3]. Ice forms as a result of water droplet impact on the airplane surfaces while flying in weather conditions with low temperature and high water droplets density. There are three major ice formations types (Figure 1.1): rime, glaze and mixed ice. Rime ice forms when all water impinging on the surface freezes instantly. It is white or opaque and usually forms in conditions with low speed, low water content and low temperatures. Wet glaze ice forms when a portion of impinging water freezes on impact while the other part flows as water on the downstream surface and might freezes downstream. It is glassy ice and usually occurs in flight conditions with high speed, high liquid water content and temperatures close to the freezing point. Mixed ice forms when flight conditions change the way that both rime and glaze ice accumulate and mix on the airfoil. Note that glaze ice has a more complex shape including ice horns compared to rime or mixed ice, and causes higher flow disturbance and larger airplane aerodynamic performance degradation [2].

Temperature range for aircraft icing accretion phenomena is approximately between -40 °C to 0 °C. The altitude range for icing phenomena is 300 to 30,000 feet, when aircraft flies in dense clouds with high water droplets content. Two basic cloud types can be categorized for icing phenomena: large horizontal extended stratiform clouds with low liquid water content and low horizontal extended cumuliform clouds with high Liquid Water Content (LWC). The main parameters affecting the ice accretion process are the environment liquid water content, droplet size, air temperature, air speed and aircraft surface roughness [3].

The icing research topics are generally categorized as, amongst others, multi-physic/multi-phase modeling, numerical simulation, icing effects and experimental research. They are all essential as they give the capability for industry to improve the design of ice accretion systems; to reduce in-flight or wind tunnel experimental icing test costs; and also to increase its ability for introducing safer airplanes to the marketplace [4-7]. The present work aims to improve the numerical simulation of various icing phenomena.

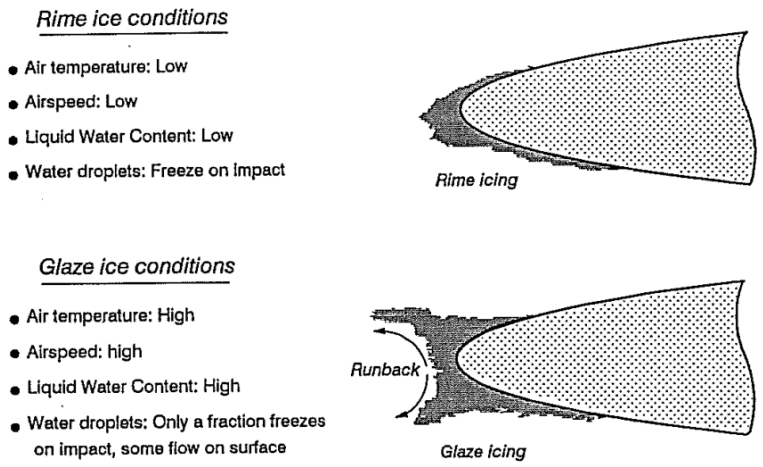

Figure 1.1: Rime and glaze ice shapes [2].

1.2 Simulation

Traditional icing simulation methods are mainly based on potential panel method flow simulation along with viscous boundary layer calculation [8-10]. The resulted potential flow solutions are used in Lagrangian droplet trajectory and impingement efficiency calculations, while boundary layer solutions provide data to the ice accretion thermodynamic model combining semi-empirical models, surface roughness effect and the heat transfer determination (also referred as Messinger model) [11]. To simplify the unsteady physic of icing problem, icing simulation is done in a quasi-steady method (single or multi-time steps), where each icing time steps is solved as a steady state problem [9]. After each time steps, the new surface is updated using the accumulated ice height on each panel and the simulation process repeated until the total icing time is reached. Using this methodology, a number of well known icing simulation codes have been developed such as LEWICE (NASA Glenn Research Center) [8, 12], CANICE (Polytechnique Montreal) [9, 13], FENSAP-ICE (Mcgill University) [14], ONERA [10]. Table 1.1 shows the specifications of a number of well known icing codes.

The icing simulation codes CANICE2D&3D have been developed at Polytechnique Montréal as part of a collaborative R&D activity funded by Bombardier Aerospace and NSERC [15-19]. It should be noted that the version presented here slightly differs from the version used at Bombardier (named CANICE-BA) and the results and conclusions presented here do not apply to CANICE-BA. CANICE2D [19] contains an inviscid flow solver (panel method) and icing/anti-icing resolution modules. The potential flow solution is used to determine the water

droplet trajectory and droplet impingement distribution via the Lagrangian approach. An integral boundary layer formulation is implemented in CANICE to determine the local heat transfer coefficient, skin friction, and near-body flow characteristics. The traditional Messinger model is used for ice accretion thermodynamic analysis. The thermodynamic model incorporates roughness, runback and water splash/ice shed models based on a water-bead model [17]. The ice shape and the amount of runback water are determined from the thermodynamic analysis. CANICE2D panel-method based is validated through NATO/RTO exercises (Figure 1.2) [9].

Table 1.1: Icing codes info.

Code	Developer	Solver Specifications			
		Mesh	**Flow**	**Droplet**	**Ice**
LEWICE (2D/3D)	NASA	-	Panel-method/BL	Lagrangian method	Traditional Messinger
		ICEG2D (Struct./Parabolic)	RANS NPARC	Lagrangian method	Traditional Messinger
CANICE (2D/3D)	Polytech. Montréal, Bombardier Aerospace	-	Panel-method/BL	Lagrangian method	Traditional Messinger
		NSGRID (Struct./ Elliptic Blended)	RANS (PMB /NSMB /NSCODE)	Lagrangian /Eulerian method	Traditional Messinger
ONERA (2D/3D)	ONERA	-	Panel-method/BL	Lagrangian method	Traditional Messinger
FENSAP-ICE (2D/3D)	McGill University	OptiGrid (Unstructured)	RANS FENSAP	Eulerian method	SWIM

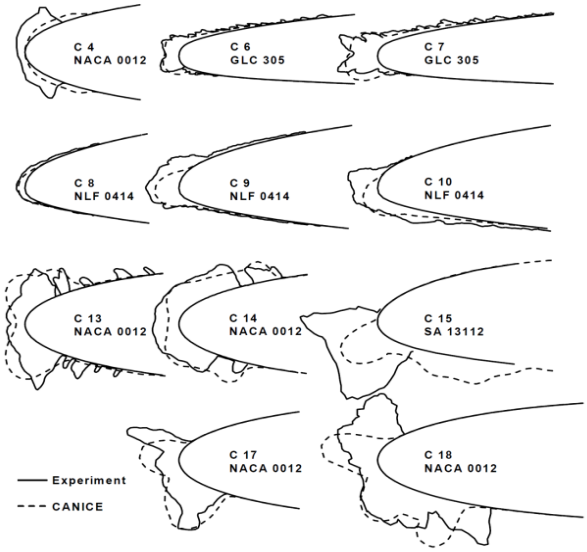

Figure 1.2: CANICE2D results and comparison for NATO/RTO exercise case studies [9].

While these models provide engineering accuracy on smooth ice shapes (i.e. rime), discrepancies occur on horn shapes (i.e. glaze) when viscosity effects play an important role. This issue prompts to move toward use of Reynolds Averaged Navier-Stokes (RANS) based flow solver [4, 12, 20]. Moving toward RANS based flow field simulation give rise to a number of issues:

1) Grid generation: Structured grids, which are the framework of this project, are considered efficient from the point of solution accuracy and computation time, but highly depend on the quality of the grid [21-24]. The proposed approaches in structured grid generation using PDE equation system with inclusion of grids control functions (stretching, orthogonality, curvature, etc.) and surface point distribution control, have improved the efficiency and quality of the grid generation process for 2D and 3D complex ice shapes domains [17]. There have been a number of grid generation tools developed mostly for iced airfoil analysis, such as SmaggIce (NASA Glenn) [25], parabolic structured and semi-structured grid generator ICEG2D (Thompson, [12]), but still they require extensive user know-how and lack multiblock compatibility. They also have poor grid metrics (orthogonality, smoothness, skewness) on complex shapes.
2) Roughness effects: A number of models have been developed to impose the surface roughness effects in RANS based flow simulation. Models such as Boeing and ONERA rough wall treatment extension are implemented in Spalart-Allmaras turbulence model [26, 27]. Models such as Wilcox and Knopp are implemented in k-ω turbulence model [26, 28]. The roughness effects also have been imposed to the calculation of heat transfer coefficient for turbulent boundary layer using a roughness based semi-empirical model. Roughness is computed using empirical sand-grain method and assumed to be constant on the surface [2]. Also a number of methodologies (based on surface water bed accumulation) are proposed to compute a non-uniform roughness on the ice surface [29].
3) Numerical methods: Computation time can be reduced through the use of acceleration methods such as multi-grid, etc. and through the use of parallel computing, which is almost a requirement when targeting 3D icing simulation process [30-32].
4) Droplet trajectory: The Lagrangian approach is used in most icing simulation tools. The Lagrangian method is based on equation of motion to compute the water droplets trajectories [5]. The method has some drawbacks such as droplet trajectory lost and

extensive computation time in 3D complex domains. A breakthrough was made by (Beaugendre [33]), who introduced an Eulerian formulation for the droplet density.

5) Ice accretion: Thermodynamic Messinger model has been used extensively for 2D and 3D ice accretion process [11, 17]. The methodology is based on simplified runback water model and single stagnation point assumption. Multiple stagnation points can be treated with an iterative Messinger model [34], while a Shallow Water Model (SWIM, [35]) provides a more accurate water runback model.
6) Unsteady flow: The unsteady problem is solved in a quasi-steady process, as decoupling the aerodynamic and ice growth effects. Recently, a fully coupled approach [36] has shown the importance of modeling the disciplines influences for complex glaze ice conditions.

Two well known CFD based icing codes are briefly introduced in next section.

1.2.1 LEWICE-NS

LEWICE-NS is one of the first ice accretion codes integrated with RANS based flow solver to perform multi-time step ice accretion simulations [8, 12]. The main difficulty of the RANS based multi-time steps icing is the complexity of grid generation for difficult geometries (ice horns or multi-element airfoils), and the deformation/regeneration/automation of the ice shape grids during the multiple time steps icing process. LEWICE is based on potential flow integrated with viscous boundary layer methods. The droplet trajectory and collection efficiency computation are based on the Lagrangian method. In RANS based LEWICE (LEWICE-NS), a parabolic grid generation code ICEG2D developed by Thompson and Sony [12] has been used to integrate the NPARC RANS flow solver to the LEWICE icing module (instead of the panel-method/integral boundary layer method). The structured grid generator is based on parabolic marching scheme, which mainly includes two steps: algebraic generation of reference grids, and smoothing using elliptic Poisson equation. To update the ice grids after each ice accretion time steps, grid points are displaced using mesh deforming functions.

The flow solution with the NPARC solver is integrated with icing module in two levels: level-1, integration of RANS flow solver results to the trajectory and impingent efficiency simulation (instead of the potential flow solution); level-2, which includes level-1 with

additional use of RANS information, namely the heat transfer parameters to the icing thermodynamic module (instead of the boundary layer method solution). The results of LEWICE-NS using the ICEG2D grid generation package for two case studies (rime and glaze ice) are shown in Figure 1.3. It is observed that the quality of the grid is not well preserved for the glaze ice case. Grid shocks, clustering/opening, smoothness and stretching are issues arising from simple parabolic method. The surface roughness effect has not been addressed in these results although it is an important criterion through the process of icing simulation as it influences the computed surface skin friction and heat transfer and results changes in ice accretion calculation [37].

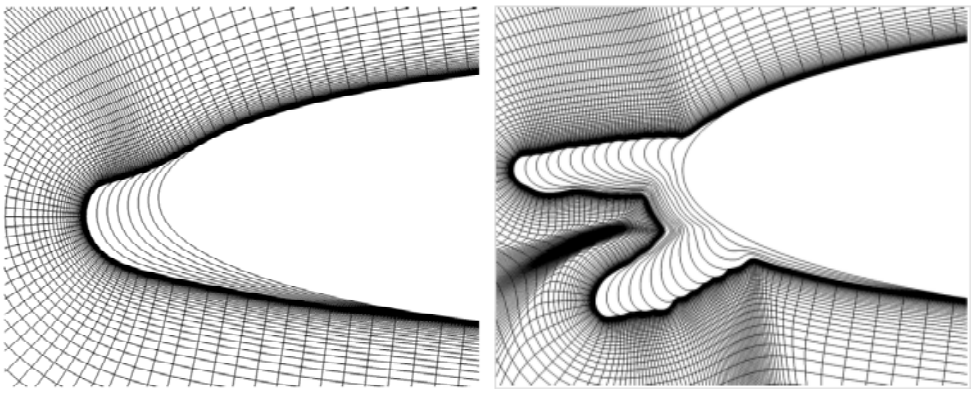

Figure 1.3: Grid generated (rime ice [left] and glaze ice [right]) [12].

1.2.2 FENSAP-ICE

FENSAP-ICE is one of the first icing simulation codes developed within an Eulerian approach [36-38]. The code includes five main modules: an unstructured grid generation module, a finite element RANS based solver (FENSAP), an Eulerian based water droplet approach, a 3D ice accretion module, and Conjugate Heat Transfer (CHT) computation in presence of anti-icing heat transfer through the wing skin. The code uses unstructured or hybrid grids and applies grid deformation for multi-time step icing simulation (Figure 1.4) [14]. Uses of unstructured grids allow treatment of more complex geometries compared to structured based methods. Roughness effect has been studied by FENSAP-ICE through the use of rough wall treatment Boeing extension implemented in the Spalart-Allmaras turbulence module [37]. A Shallow Water Icing Model (SWIM) model is developed for the ice accretion and water runback computations, to complete the Messinger model. The code includes a hot air jet anti-icing module incorporated

with the thermodynamic computation through CHT module. The third generation of FENSAP-ICE-Unsteady has been developed to perform unsteady icing simulation [36].

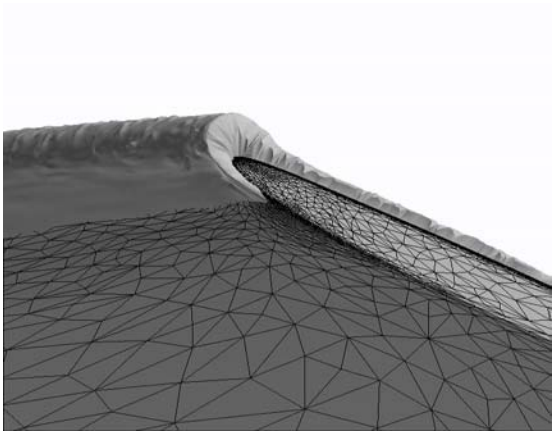

Figure 1.4: FENSAP-ICE 3D ice solution [14].

1.3 Aerodynamic effects

It is known that an iced airfoil shows significant reduced lift and increased drag. This consequently results in stall at lower angle of attack compared to clean airfoils as well as decrease aircraft speeds [6, 39]. Glaze ice forms create a more destructive effect with large flow separation and early stall compared to rime ice forms. Generally, there are three ways to determine the icing effects on the airplane aerodynamic performances: 1) Flight testing; 2) Wind tunnel testing; 3) CFD simulations. These assessment methods differ in cost, simulation constraints for realistic test conditions, and modeling errors. Flight tests are very expensive, but they are the best way to have the most realistic conditions and results, although freestream conditions are not easy to characterize [4]. Wind-tunnel tests are less expensive albeit still out of reach of routine use, but they give additional flexibility to change and control tests conditions and icing parameters [7]. One drawback is the geometry scaling factor which significantly alters the results. The cheapest method is the application of CFD to model in-flight icing with a high flexibility to control the icing and flow conditions [6]. The accuracy of the CFD approaches depend on mesh quality, efficient turbulence models, boundary layer transition models, steady or unsteady simulations, etc. However, they can model flight test Reynolds number while aiding extrapolation of wind-tunnel results.

CFD methods help to determine proper trends with respect to the various parameters. RANS flow simulations can predict the flow behavior over simple ice shapes but lack sufficient

accuracy when ice shapes are more complex, such as those with large ice horns and very rough surfaces, or with unsteadiness effects. In these cases, there can be significant differences between the numerical predictions and the experimental results. In CFD analysis, grid sensitivity studies can be helpful to find the optimal grid properties for computation of the flow around the iced airfoil. The effects of grid quality and density can play a vital role when the analyses involve cases of high angle of attack or flow separation [40]. CFD methods need verification, as well as validation using experimental data. One of the major difficulties in validation of icing simulation codes is the availability of precise experimental data bases.

Measuring the ice shape by 2D cross sections has been so far as the most prefer method to obtain the experimental ice shapes data. Measuring ice shapes is very difficult because the ice surface includes large number of sharp edges, feathers, porosity and small size roughness. Also the ice deformation and melting can occur during the measurements. Covering the ice with paint or powders and using optical scanning is another method for measuring the ice shape, but it still has many difficulties and measurement errors [4, 7].

1.4 Thesis objectives

In view of the context described above, the overall goal of the work is to advance the numerical modeling towards advancing our understanding of the quasi-steady ice accretion process for complex ice accretion and aerodynamic effects.

In particular, the specific objectives are:

1- Examine the impact of RANS flow solver on the ice accretion framework.

2- Examine multi-steps ice accretion quasi-steady convergence.

3- Develop novel grid generation algorithms specifically for ice shape/growth.

Note that the research is centered around algorithms development, thus excluding studies of specific physical phenomena such as Supercooled Large Droplets (SLD) or non-uniform surface roughness effects. Also, although the framework is linked to both Lagrangian and Eulerian droplet formulations, only the Lagrangian approach is used in the thesis.

Towards the thesis objectives, the following platforms are used:

1) Polytechnique Montreal (PM) icing code, CANICE2D [15], based on Lagrangian formulation.
2) Polytechnique Montreal (PM) grid generation code, NSGRID2D&3D [41], (developed as a part of this research project).
3) ICEM-CFD commercial grid generation package [42].
4) PMB3D RANS solver [43], University of Liverpool.
5) NSMB3D RANS solver [44], CFS Engineering Inc.
6) Polytechnique Montreal (PM) RANS solver, NSCODE2D [45].

The CANICE2D framework provides the starting point, with RANS capabilities inserted gradually in a sequential manner to finally obtain the CANICE2D-NS framework [20, 39, 46]. The developed framework then is used to generate a validation database and enable studies of icing effects on aircraft aerodynamics.

Note that the developments in NSCODE2D-ICE [34], NSGRID3D and icing framework of NSMB3D-ICE/NSGRID3D, published in ref. [47], are excluded from the content of this thesis. The thesis focuses on the development of two dimensional grid generation code NSGRID2D and RANS aero-icing framework CANICE2D-NS. The developed codes specifications are shown in Table 1.2.

Table 1.2: PM icing codes info.

Code	**Developer**	**Solver Specifications**			
		Mesh	**Flow**	**Droplet**	**Ice**
CANICE-NS (2D) Multi-steps	PM	NSGRID (Structured/ Elliptic Blended)	RANS (PMB /NSMB /NSCODE)	Lgrangian /Eulerian method	Traditional Messinger
NSCODE-ICE (2D) Multi-steps	PM	NSGRID (Structured/ Elliptic Blended)	RANS NSCODE	Eulerian method	Iterative Messinger
NSMB-ICE (2D/3D) Multi-steps	PM/ Uni. de Strasbourg	NSGRID (Structured/ Elliptic Blended)	RANS NSMB	Eulerian method	Iterative Messinger

The thesis is organized as follows. Chapter 2 addresses the literature review, icing methodologies and algorithm developments. Chapter 3 addresses the consistency of the works and the publications toward the objectives of the project. Chapters 4 to 6 include the main publications, developments and discussion that address the three objectives of the thesis. A general discussion is presented in Chapter 7, before a conclusion in Chapter 8. Appendix A includes additional modeling descriptions. Appendix B includes additional NSGRID2D and CANICE2D-NS results and discussions.

CHAPTER 2

LITERATURE REVIEW AND METHODOLOGY

First the literature review is presented which includes: background in icing problem, the detailed description of traditional ice accretion process, and grid generation tools/methods survey for ice mesh generation problem. Finally the solution methodologies are addressed.

2.1 Background

2.1.1 Context

To reduce the number of incidents/accidents due to icing and enlarge aircraft certificates for flying in all weather conditions, one needs to develop icing simulation capabilities. Prediction of precise complex ice accretion resulting in aircraft performance degradation represents a challenge in aeronautical science. The National Transport Safety Board identified ice accretion and its effects as one of the major causes of flight accidents [1]. Ice forms on different surfaces of the aircraft, when flying in weather conditions with temperature lower or close to freezing point and with water droplets impaction on the aircraft surfaces. Ice can form with different characteristics categorized generally in three types of formations: rime, glaze, and mixed ice. The physical complexity of ice formations on glaze ice is higher than for rime ice. For instant, ice horns growth can lead to very odd shapes that cause higher flow disturbance and aerodynamic performance degradation [3].

In the field of ice accretion simulation, the traditional ice prediction simulation packages, such as LEWICE (NASA Glenn Research Center) [48], CANICE (Polytechnique Montreal) [18], are mainly based on potential flow solvers coupled with two dimensional boundary layer methods. The flow field solution is used to compute the water droplet trajectories and water collection efficiency on the aircraft via a Lagrangian approach, and also to compute the heat transfer coefficient needed for the ice accretion thermodynamic module [17]. Knowing the incoming water mass accumulation rate, water runback and heat transfer properties, the amount of accumulated ice is determined for a specific time period. Finally the surface geometry is updated using the amount of ice formed on the surface. Since, growth of ice on the body influences the flow field [31], the total ice accretion simulation is obtained by breaking the total icing time to

specific number of time steps, where each step is treated as a steady state flow problem (this approach is referred as multi-step procedure, using a quasi-steady approximation with each step).

CANICE is an ice accretion simulation code developed at Polytechnique Montreal under research funded by NSERC and Bombardier Aerospace. The code includes four basic modules: external flow simulation, droplet trajectory and local catch efficiently calculation, surface ice/water interface thermodynamic balance and ice accretion and hot air anti-icing simulation (Figure 2.1).

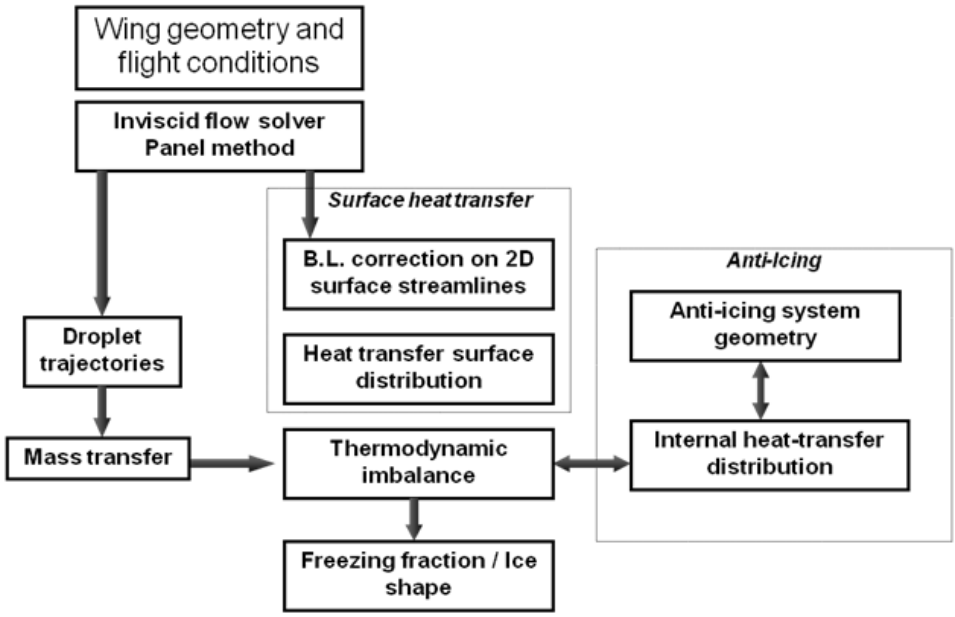

Figure 2.1: CANICE code structure [2].

The CANICE2D code can perform for multi-element, multi-time steps icing simulations of airfoils. The flow solution is obtained by the Hess and Smith panel-method approach and is used for droplet trajectory and viscous boundary layer calculation. CANICE3D is the extension of CANICE2D to three dimensions, which has been integrated to CMARC, a low order three-dimensional panel method code, to compute the potential flow solution around the wing needed for the 3D droplet trajectories and impinging efficiency calculations [49, 50]. For both codes, a decoupled integral boundary layer or coupled viscous-inviscid interaction method is used to simulate viscous effects. To include the effect of roughness in the boundary layer model, the transition criteria, skin fiction and heat transfer calculation for laminar and turbulent region have been modified using roughness based empirical equations. Roughness is based on the equivalent sand-grain roughness height, which is calculated with an empirical model [15].

Droplet trajectory is simulated using the Lagrangian approach, solving the equation of motion of the water droplets for the defined time intervals using a forth-order Runge-Kutta scheme. The forces acting are the drag force on the spherical water droplets along with gravity and buoyancy force. The module thus determines the water droplet trajectories and impingement on the surface [49]. The droplet impingement distribution defines the water droplet local catch efficiency on the surface which determines the droplet mass flow rate impinged on the surface panels. Using the calculated local convective heat transfer and water droplet impingement flow rate on the surface panels along with runback water mass rate from the neighbor panels, the mass and energy balance is applied to the surface control volumes to calculate the amount of supercooled impinged water droplets converted to ice mass. The Messinger model is used to define the type of ice surface (wet, dry rime or wet glaze), freezing fraction (fraction of ice mass to the entering total mass flux to the control volume), and surface temperature [2, 11]. To take into account the hot air anti-icing heat flux, the anti-icing module solves the internal heat transfer coefficient using an empirical correlation related to impinging jet on flat plate. The correlation takes into account the average Nusselt number based on the jet parameters such as jet Reynolds number, nozzle to surface distance, and nozzle width [16]. Using the calculated internal hot-air local heat transfer coefficient and conduction through the thin leading edge skin, in an iterative process, wall heat transfer rate is calculated and integrated to the surface control volume thermodynamic balance. Final ice geometry is updated by calculating the ice height growth at the center of each panel and interpolating and smoothing the panel's connectivity on the edges using panel size and angle criteria [51]. This step is not as easy as it appears, so a non-conservative method is used to simplify the process.

The panel method flow solver shows increasing errors in the flow solution as the complexity of the ice shapes increases, such as glaze ice with horns. The errors mainly comes from the higher influence of viscous effect and shows the need for RANS based flow solution for icing phenomena simulation and effects analysis [52]. The main difficulty of CFD based icing simulation comes from the point of grid generation for complex ice shapes, computation time and memory usage. New improvement in CFD based flow solution such as modified turbulence models with roughness effects, solution approaches for speeding up the convergence, parallel computing, unsteady flow solution, along with new developments in grid generation, smoothing and automation methods, have increased the efficiency of CFD based approaches in the ice

accretion simulation and effects analysis [53]. Note that CFD methods belonging to the class of Immersed Boundary Method (IBM), which considerably eases the grid generation problems of body-conforming methods, are not addressed here.

Grid generation methods, such as structured, unstructured, and hybrid grids, have been applied to icing problems [6, 14]. A finding of the NASA "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences" explicitly says "Mesh generation and adaptivity continue to be significant bottlenecks in the CFD workflow, and very little government investment has been targeted in these areas" [54], perhaps explaining the few developments in multi-block structured mesh generation algorithms in the last decades. Structured-grid methods suffer from the aspect of flexibility and grid quality for complex shapes such as ice accretion, but the flow solver is typically more efficient and accurate. The main difficulty of structured mesh in complex ice shapes domains is related to the grid smoothing approaches on physical domain boundaries which contain discontinuities, sharp angles, highly concave and convex area, etc. The process of grid generation for flow simulation on iced airfoils typically includes three basic steps: determination and smoothing of the iced surface; distribution of the grid points on the surface; and finally generation of the volume mesh using various approaches [23, 55]. There are three major classes for structured grid generation: algebraic methods, partial differential equation (PDE) methods, conformal mapping methods. PDE methods can be categorized as elliptic, hyperbolic and parabolic methods [23, 24].

PDE based methods such as elliptic Poisson equation (based on Laplacian operators), are solved by finite difference or finite volume discretizations to generate smoothed structured grids. They are more complex than algebraic methods but provide more flexibility to generate high quality grids using control functions. These methods use an algebraic grid generated as a starting grid (initial solution) and perform an iterative procedure to generate the desirable grid quality [22, 55]. There are still some difficulties using Laplacian operators, such as controlling grid spacing close to the wall or generating large cells in concave domain and small cells in convex regions. Control functions such as mesh spacing control, orthogonality control, and curvature control have been developed and used to increase the control on the generated grid quality. Spacing control functions modify the wall normal spacing and the stretching ratio of the grids. Orthogonality control functions modify the grid orthogonality and skewness throughout the domain. Curvature control functions modify normal spacing in concave/convex regions and grid

negative volumes [22, 56]. These grid controls, especially close to the wall for viscous dominated regions, have extensive influence on Navier-Stokes flow solution accuracy. It is thus important to have grids with good metrics, which are very difficult to obtain when geometries contain concave/convex domains with sharp corners [57]. A number of well known control functions developed by Sorenson, Steger, Thompson, Soni, and others, solve for spacing and orthogonality on the boundary and propagate the information throughout the domains using Laplacian operators [21, 23, 55]. Other control functions developed by Spekreije, solve the spacing, orthogonality and curvature using parametric space and impose the properties to the physical grid domain to retain the desired quality [22]. There are also methods to control grid quality such as boundary orthogonality approaches developed by Khamayseh, Kuprat through the use of specified Dirichtlet or Numann boundary condition in elliptic equation system of grids [23]. For the problem of ice, no elliptic method has yet produce adequate metrics, because of the complexity of the ice surface.

Conformal mapping approaches such as parabolic marching methods are another grid generation class discussed for the problem of ice grid generation. Thompson and Soni have developed an efficient parabolic grid generation code ICEG2D integrated to LEWICE and NPARC solver to perform multi-time steps ice accretion simulation [58]. The grid generation code is capable of surface and field grid generation and smoothing. The surface grid generation is done using a weight functioning based on the surface points. Also the surface is smoothed, to remove the sharp point complexity, prior using the parabolic approach. Surface is defined by NURBS functions. To develop the field grid, the code uses a conformal mapping approach and generates structured and semi-structured grids. Semi structured grids include quad and hexahedral elements. The structured grid approach includes two main steps: algebraically generation of locally orthogonal reference mesh and smoothing of the reference mesh using Poisson equation. To generate the semi-structured grid one need to add a third step: deletion and insertion algorithm based on the reference grid quality and defining the appropriate initial data surface for next layer generation. The efficiency of the method is addressed for icing aerodynamic performance simulation and analysis in reference [59]. Difficulties of structured parabolic grid rise from grid clustering and grid opening caused by concave and convex domains, respectively, and their propagation throughout the domain. The other issue is the nature of parabolic methods that are not well applicable for multi surface grid. In semi-structured grids,

the memory usage and the flow solver developments represent the main difficulties. These grids have unstructured data storage method and applications.

Some approaches and tools have also been introduced specifically for structured ice grid generation, and are mainly based on heuristics approaches. Shih developed an interactive tool to generate multi-block structured algebraic grid and block modification for ice grid problem [6]. The tool includes a large number of grid manipulation and modification applications on geometry, blocks, edges, domain grids, etc. The code also includes simple elliptic domain smoothing. To prepare the ice surface grid, first the ice geometry is smoothed to remove the complexity of the ice shapes with horns, sharp edges, feathers and surface roughness. Then to generate the field grid, one needs to define the blocking topology as single block or multi-block. Both blocking topologies result in some advantages and disadvantages to control the grid quality. Single block method includes two domain sections, a fine grid domain close to the ice shape that will be smoothed and a coarse grid domain in far field. The multi-block method includes a thick wrap-around section covering the ice shapes that minimizes the influence of ice geometry on the field grid. The rest of the domain is divided into as many numbers of blocks for better grid manipulation. Both single and multi-block methodologies are very time consuming and need extensive application by user. Automation of the methodologies is also an issue. The generated grids by these approaches have been used for different application of iced airfoil RANS based flow simulations.

Although there have been many proposed grid methods, CFD applications for icing simulations remain impeded by the absence of a robust grid generation process that results in high quality Navier-Stokes resolution grids for severe mixed concave/convex glaze ice shapes. The process of mesh generation in multi-time steps icing sequences can be performed via mesh regeneration or mesh movement (spring analogy [60], elastic [61], adjoint-based [30]). Although mesh movement is used by some researchers for icing simulations [58, 38], it has been found that, at least in 2D, the alternate approach of mesh regeneration provides increased robustness [13]. Automation of the grid generation process is another important aspect for performing multi-time steps icing simulations.

The other important aspect in CFD based icing simulation is the modeling and imposing the effect of roughness in ice accretion process as it highly influences the computed surface skin friction and heat transfer rate that results in changes in ice accretion determination [31, 37].

Surface roughness height generally is computed using empirical models such as sand-grain model [2] within boundary layer codes. To impose the effect of roughness for RANS based icing simulation, one need to implement the roughness effects in standard turbulence models. For this aspect, the rough wall treatment implementation in turbulence models such Spalart-Allmaras and k-ω models have been developed [26]. For the Spalart-Allmaras turbulence model, there are two extensions: the ONERA and Boeing models [27]. The roughness is incorporated to the S-A model by affecting the turbulent eddy viscosity in the wall region. One can relate the roughness height to the changes in velocity profile which changes the wall skin friction. Using two equations k-ω model, two rough wall models have been addressed: Wilcox method and Knopp method [28].

Generally the roughness value is assumed to be constant on the ice surface but in reality its value changes through time and space. There have been attempts to compute its value based on the water beading on the surface [62]. Based on the observed behavior of the impinging water on the surface, water droplets at first form small beads with spherical shape (because of the surface tension force). The beads start to grow as more droplets impinge the surface. Part of the bead freezes while the other liquid part covers the surface. As the beads grow in size, it finally reaches a maximum bead height and the water starts to flow on the surface due to the air flow shear force. The frozen part of the bead (ice bead) forms the surface roughness height which is taken equivalent as the sand grain roughness height and used in single and multi-time steps icing simulation [2, 63].

For the problem of icing droplet trajectory and impinging efficiency calculation, two approaches have been addressed: Lagrangian and Eulerian methods. In the traditional Lagrangian approach where the droplets are treated, the equations of motion for the droplet are solved to compute the droplet velocity and position in space through the time intervals [5]. The difficulties of the Lagrangian approach are its high computation time and collection efficiency calculation in complex domains. In the Eulerian approach, the droplet flow field is solved as a two-phase flow problem through the conservation mass and momentum equations [32, 33]. Solving the continuity and momentum equation for the water droplets provides droplets velocities and water volume fractions solution in the domain. The water droplets velocity is solved on the same mesh and nodes than the fluid and the water flux impinging on the surface is calculated directly [49, 64]. Different approaches such as FEM or FVM can be applied for discretization of the

hyperbolic systems [65, 66]. One of the main advantages of the Eulerian approach is that the discrete droplet trajectory as the Lagrangian approach is not solved and therefore it reduces the computational memory usage and cost time especially in three-dimensional trajectory simulation. Also the Eulerian collection efficiency can be used simply for multi-element airfoil without the difficulty of impinging boundary calculation difficulties presence in Lagrangian approach [34].

Another important aspect is the modeling of the physics and effects of Supercooled Large Droplet (SLD) in ice accretion simulation. Supercooled large droplet icing condition (i.e. 40 μm to 400 μm diameters) have been addressed by the Federal Aviation Administration as one of the main concerns for ice accretion simulation codes to increase flight safety [1]. To simulate the SLD effects in icing simulation codes, SLD deformation, break up, splashing and rebounding need to be studied and modeled. SLD deformation make changes in droplet drag force and its trajectories. SLD break up generates secondary droplets which need to be tracked again and their trajectory calculated as well. SLD splash or rebounding also influences water catch efficiency and impinging water loss [67, 68]. A number of empirical models have been developed to modify the calculation of the droplet trajectory and collection efficiency in Lagrangian or Eulerian methods [69]. The study of SLD models is outside the scope of this research.

Ice accretion thermodynamics performs a mass and energy balance on the impinging surface. To calculate the accreted ice mass on the surface from the impinging droplet velocity and impinging water mass, and water runback, the surface heat flux and convective heat transfer coefficient are used to solve the governing equations on the surface control volume [2, 8]. The traditional Messinger model is used extensively for the ice accretion calculations, which along with the thermodynamic equations, defines the surface type, freezing fraction and surface temperature [11, 70]. Using the traditional method, one needs to have a single stagnation point to be capable of defining the runback water directions. An iterative Messinger model is developed to improve the difficulty of multi-stagnation problem and water runback computation [32, 34]. The iterative Messinger model also improves the limitation if extend to 3D equation system. Ice accretion can also be modeled via partial differential equation systems. Based on this model, mass, momentum and energy conservation PDE equation systems are solved to compute the surface water film flow and ice height thickness. This model is named Shallow-Water Icing Model (SWIM) and is used in FENSAP-ICE [35]. Transformation of the thermodynamic

Messinger model to a partial differential model also improves the ability to extend from 2D model to 3D systems.

Finally, numerical efficiency must be addressed when simulating multi-step icing problem in 2 and 3 dimensional domain. This area has received less attention and is extensively examined in the present work.

A more detailed description on the proposed approaches and methodologies in ice accretion simulation process are described in the next sections.

2.1.2 Traditional ice accretion framework of CANICE

Here we present the detailed modules of the traditional icing simulation code CANICE. The icing simulation process includes three main modules: 1) Panel method flow simulation, 2) Lagrangian droplet computation, 3) Traditional Messinger model thermodynamics and ice accretion.

2.1.2.1 Panel method flow simulation

Flow field parameter solution for the icing simulation in CANICE code is obtained by potential inviscid flow integrated with viscous boundary layer calculation [2, 8, 14]. The flow is assumed inviscid, irrotational and incompressible. The continuity and momentum equation for inviscid flow are [2, 53]:

$$\nabla.\vec{V} + \frac{\vec{V}}{\rho}.\nabla\rho = 0 \tag{2.1}$$

$$\vec{V}.\nabla\vec{V} = -\frac{\nabla \mathrm{p}}{\rho} \tag{2.2}$$

where $\vec{V}$ is the velocity, p is the pressure and ρ is the density. For incompressible (ρ constant) and irrotational flow ($\nabla \times \vec{V} = 0$), the continuity and momentum equation simplify to:

$$\nabla.\vec{V} = 0 \tag{2.3}$$

$$\nabla(\frac{|\vec{V}|^2}{2}) = -\frac{\nabla \mathrm{p}}{\rho} \tag{2.4}$$

Using the velocity potential Φ defined as $\vec{V} = \nabla\Phi$, the continuity equation becomes $\nabla^2\Phi = 0$. The Hess and Smith panel-method is used as the flow solver of the Laplace equation, which is a linear partial differential equation of the second order. More complex potential flows can be simulated by the sum of elementary flow potentials Φ_e and superimposing singularities such as sources, vortices and doublets:

$$\Phi = \sum_{e=1}^{n} \alpha_e \Phi_e \tag{2.5}$$

These singularities are distributed on the body surface, discretized by a number of panels. The linear equations system is solved to compute the flow parameters. The potential at the point P is written as:

$$\Phi(P) = \Phi_\infty(P) + \int_{S_c} q(s)\, \Phi_s ds + \int_{S_c} \gamma(s)\, \Phi_v ds \tag{2.6}$$

where Φ_∞ is the free stream potential, Φ_s is the potential of source distribution of strength $q(s)$ per unit length, and Φ_v is the potential of vortex distribution of strength $\gamma(s)$ per unit length. The discretization of equation for N panel reduces to:

$$\Phi(P) = \Phi_\infty(P) + \sum_{j=1}^{N} q_j \int_0^{l_j} \Phi_s ds + \sum_{j=1}^{N} \gamma_j \int_0^{l_j} \Phi_v ds \tag{2.7}$$

The $q_j(s)$ source strength is assumed to be constant for the panel j, and γ_j vorticity strength assumed to be constant on all the panels. The velocity $\vec{V}$ at each point (x, y) is:

$$\vec{V} = \vec{U} + \vec{v} \tag{2.8}$$

where $\vec{U}$ is the uniform velocity at the infinity ($\vec{U} = V_\infty(cos\alpha\vec{\imath} + sin\alpha\vec{\jmath})$), and $\vec{v}$ is the perturbation flow velocity due to the body which has been represented by source and vortex flows. The normal and tangential velocity components of the total velocity at each panel control point i can be written as:

$$V_i^n = \sum_{j=1}^{N} A_{i,j}^n q_j + \sum_{j=1}^{N} B_{i,j}^n \gamma_j + V_\infty \sin(\alpha - \theta_i) \tag{2.9}$$

$$V_i^t = \sum_{j=1}^{N} A_{i,j}^t q_j + \sum_{j=1}^{N} B_{i,j}^t \gamma_j + V_\infty \cos(\alpha - \theta_i) \tag{2.10}$$

where $A_{i,j}^n$, $A_{i,j}^t$, $B_{i,j}^n$, $B_{i,j}^t$ are the influence coefficient related to the airfoil geometry [53].

Using the Kutta-Joukowski condition for trailing edge tangential velocities ($V_n^t = -V_1^t$), normal velocity condition ($V_i^n = 0$) for each panels, and γ_i vorticity strength assuming to be constant as γ, the normal velocity V_i^n equation reduces to:

$$\sum_{j=1}^{N} A_{i,j}^n q_j + \sum_{j=1}^{N} \gamma B_{i,j}^n + V_\infty \sin(\alpha - \theta_i) = 0 \text{ ; for i = 1,2,…, N} \tag{2.11}$$

The final generated algebraic equations system $[A]\{x\} = \{b\}$ can be solved using methods such as the GAUSS elimination approach [2].

Integral boundary layer methods using the potential flow solution is applied for the calculation of the boundary layer properties, skin friction and heat transfer with inclusion of the surface roughness effect. Normal pressure gradient is assumed negligible for the viscous

boundary layer calculation. The laminar boundary layer can be modeled by Thwaites method [71]. The laminar momentum thickness θ_l is defined by:

$$\frac{\theta_l^2}{\nu} = \frac{0.45}{u_e^6}\int_0^s u_e^5 ds \tag{2.12}$$

where u_e is the surface velocity computed by the potential flow solver. The laminar boundary layer thickness δ is computed by:

$$\delta = \frac{315}{37}\theta_l \tag{2.13}$$

The heat transfer coefficient ($h_{c,l}$) for the laminar region is defined by:

$$h_{c,l} = 0.296\frac{k_a}{\sqrt{\nu}}[u_e^{-2.88}\int_0^s u_e^{1.88} ds]^{-0.5} \tag{2.14}$$

where k_a is the air thermal conductivity. The transition from laminar to turbulent regime is computed using the roughness based Von Doenhoff criterion:

$$R_{ek} = \frac{u_k k_s}{\nu} \tag{2.15}$$

where k_s is the roughness height, u_k is the local velocity at y = k_s :

$$\frac{u_k}{u_e} = \frac{2k_s}{\delta} - 2(\frac{k_s}{\delta})^3 + (\frac{k_s}{\delta})^4 + \frac{1}{6}\frac{\delta^2}{\nu}\frac{du_e}{ds}\frac{k_s}{\delta}(1-\frac{k_s}{\delta})^3 \tag{2.16}$$

For $R_{ek} \geq 600$, the boundary layer is assumed to be turbulent. The turbulent momentum thickness θ_t is defined by:

$$\theta_{t(s)} = [\frac{0.0156}{u_e^{4.11}}\int_{\acute{s}}^s u_e^{3.86} ds]^{0.8} + \theta_{l\acute{s}} \tag{2.17}$$

where $\acute{s}$ is the value of s at the transition point. The turbulent skin friction coefficient is computed by:

$$\frac{C_f}{2} = \frac{0.168}{[\ln(864\frac{\theta_t}{k_s})]^2} \tag{2.18}$$

The Stanton number is defined by:

$$St = \frac{C_f/2}{Pr_t + \frac{\sqrt{C_f/2}}{St_k}} \tag{2.19}$$

where $Pr_t = 0.9$ is the turbulent Prandtl number for air, and St_k is the roughness Stanton number, calculated by:

$$St_k = 1.156[\frac{u_\tau k_s}{\nu}]^{-0.2} \tag{2.20}$$

where shear velocity is calculated as ($u_\tau = u_e\sqrt{\frac{C_f}{2}}$).

The heat transfer coefficient for the turbulence region is defined by:

$$h_{c,t} = St.\rho.u_e.c_p \tag{2.21}$$

Roughness height is based on the equivalent sand-grain roughness model [2]. The resulted heat transfer coefficient is then added in the icing thermodynamic calculation.

2.1.2.2 Lagrangian droplet computation

To estimate the amount of water impinging on the surface, one way is to track the droplet trajectory from the far field to the body surface using the Lagrangian approach. The equation of motion is solved for the released droplet during the time intervals to track the droplets trajectories (Figure 2.2) [2, 49, 50, 72]. Droplets trajectories have deviations compared to the air streamlines and this deviation increases with the droplet size. The equation of motion is obtained using Newton's Law for droplets:

$$\vec{F} = m_d \frac{d^2\vec{r}_d}{dt^2} \tag{2.22}$$

The assumptions chosen:

- Volume of the droplets remain constant
- Droplets shape are not necessarily spherical
- Droplets do not affect the flow field
- Initial droplets have the farfield velocity
- Droplet density is constant
- Droplet lift is assumed to be zero

The forces are: droplet drag, gravitational and buoyancy forces:

$$m_d\vec{a}_d = \vartheta(\rho_a - \rho_d)g\left(-sin\theta\vec{\imath} + cos\theta\vec{k}\right) + \frac{1}{2}\rho_a S_F C_D\left|\vec{V} - \vec{V}_d\right|(\vec{V} - \vec{V}_d) \tag{2.23}$$

where m_d is the droplet mass, $\vec{a}_d$ is droplet acceleration, ρ_a and ρ_d are air and water droplet densities, ϑ is the water droplet volume, $\vec{V}$ and $\vec{V}_d$ are the velocities of fluid and droplet, C_D is the droplet drag coefficient, and S_F is the droplet area normal to free stream direction.

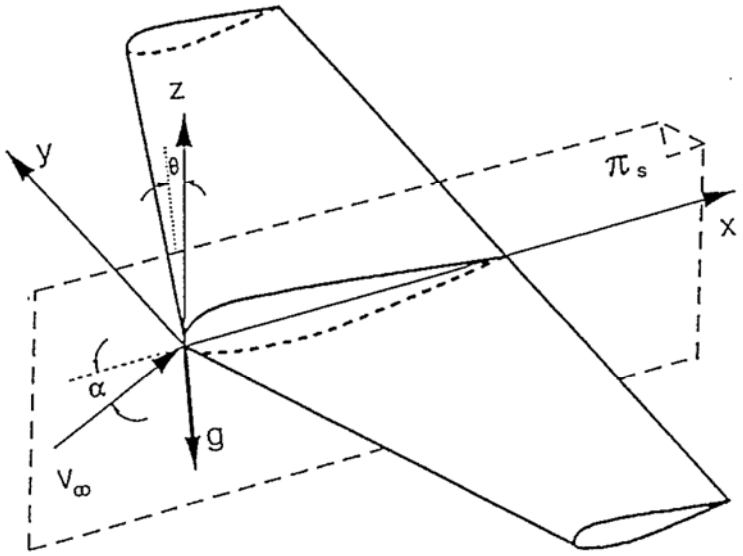

Figure 2.2: Reference system definition for droplet trajectory calculation [49].

Droplet gravity force is:

$$\vec{F}_g = \vartheta \rho_d g(sin\theta\vec{\imath} - cos\theta\vec{k}) \tag{2.24}$$

Droplet buoyancy force is:

$$\vec{F}_b = \vartheta \rho_a g(sin\theta\vec{\imath} + cos\theta\vec{k}) \tag{2.25}$$

Assuming the droplet shape as a sphere, we have:

$$m_d = \rho_d \left(\frac{3}{4}\pi\right)\left(\frac{D_{eq}}{2}\right)^3 \tag{2.26}$$

$$S_F = \frac{\pi}{4} {D_{eq}}^2 \tag{2.27}$$

$$\vartheta = \frac{\pi}{6} {D_{eq}}^3 \tag{2.28}$$

$$Re_d = \frac{\rho_a}{\mu_a} D_{eq} \left|\vec{V} - \vec{V}_d\right| \tag{2.29}$$

where D_{eq} is droplet equivolumetric diameter, and Re_d is droplet Reynolds number. Air forces are represented by pressure and shear forces on the droplet surface. It has been observed, for non rotational droplet, irrotational flow and axisymmetric droplet shape along U direction, that droplets would not have any lift forces [2]. To calculate the droplet drag force, both pressure and shear forces need to be taken into account. Droplet drag coefficient is calculated via the empirical model of Gunn and Kinzer and adapted interpolation referred in reference [49]:

For $Re \leq 1$: $C_d Re / 24 = 1$ (2.30)

For $1 \leq Re \leq 175$: $C_d Re / 24 = 1.00098596+0.26089615[\ln(Re)]-0.13329568[\ln(Re)]^2+0.10783876[\ln(Re)]^3-0.02343379[\ln(Re)]^4+0.002442409[\ln(Re)]^5$

For $175 \leq Re \leq 3500$: $C_d Re / 24 = -43396.14356345+34106.69196587[\ln(Re)]-10691.51498487[\ln(Re)]^2+1671.42366542[\ln(Re)]^3-130.35326094[\ln(Re)]^4+4.0600087[\ln(Re)]^5$

For $Re \geq 3500$: $C_d Re / 24 = 1.699\ (10^{-5})\ Re^{1.92}$

The equation of motion finally becomes:

$$\frac{d^2\vec{r}_d}{dt^2} + \frac{C_D Re}{24}\frac{1}{K_A}\frac{d\vec{r}_d}{dt} = K_G\left(sin\theta\vec{\imath} - cos\theta\vec{k}\right) + \frac{C_D Re}{24}\frac{1}{K_A}\vec{V}_a \quad (2.31)$$

$$K_A = \frac{\rho_d {D_{eq}}^2}{18\mu_a}\ ,\ K_G = \frac{(\rho_a - \rho_d)}{\rho_d}g \quad (2.32)$$

This second order differential equation can be solved using classical methods. The fourth, fifth and seventh order Runge-Kutta along with Adams-Moulton method are applied to solve the droplet equation of motion, as referred in reference [49]. Here is the position and velocity solution using the fourth order Runge-Kutta:

$$\vec{r}_{d(i+1)} = \vec{r}_{di} + \tau\vec{V}_{di} + \frac{\tau}{6}(k_1 + k_2 + k_3) \quad (2.33)$$

$$\vec{V}_{d(i+1)} = \vec{V}_{di} + \frac{\tau}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (2.34)$$

The time interval (τ) should be small enough to reduce the calculation errors while providing reasonable computation times. The new droplet position and velocity are obtained and used as initial solution for the next time interval. Initial positions of the droplet release typically are five chords upstream of the leading edge. Based on the assumptions above, initial droplet velocity has the same velocity as air velocity at this point. The complete Lagrangian trajectory calculation is implemented in CANICE2D&3D [2, 50] and CANICE2D-NS [20].

The droplet shape assumption is important for droplet drag coefficient prediction. For large droplets the shape deviates from spherical which increases the errors in drag force prediction. One of the drag models for non-spherical large droplet models is developed by Clif, Grace and Weber [64]:

$$C_D = e\ C_{D,disk} + (1\text{-}e)\ C_{D,sphere} \quad (2.35)$$

This model divides the droplet drag coefficient in a disk and a sphere and interpolates these predicted coefficients with the (e) fraction which depends on the Weber number (We):

$$e = 1 - \frac{1}{(1+0.007\sqrt{We})^6} \tag{2.36}$$

$$We = \frac{\rho V_{rel}^2 d_p}{\sigma_w} \tag{2.37}$$

The disk drag coefficient is calculated as:

$$\text{Re} \leq 0.01: \quad C_D = \frac{64}{\pi Re}\left(1 + \frac{Re}{2\pi}\right) \tag{2.38}$$

$$0.01 < \text{Re} \leq 1.5: \quad C_D = \frac{64}{\pi Re}(1 + 10^x)$$

$$1.5 < \text{Re} \leq 133: \quad C_D = \frac{64}{\pi Re}(1 + 0.138Re^{0.792})$$

$$\text{Re} > 133: \quad C_D = 1.17$$

where $x = -0.883 + 0.906 \log_{10}(\text{Re}) - 0.025 (\log_{10}(\text{Re}))^2$. After calculation of droplet trajectories, the local catch efficiency is computed based on the ratio of the area of the droplet impingements to the surface area normal to free stream velocity. The total catch efficiency is the ratio of the wing area impinging by droplets to the wing area normal to the free stream (Figure 2.3).

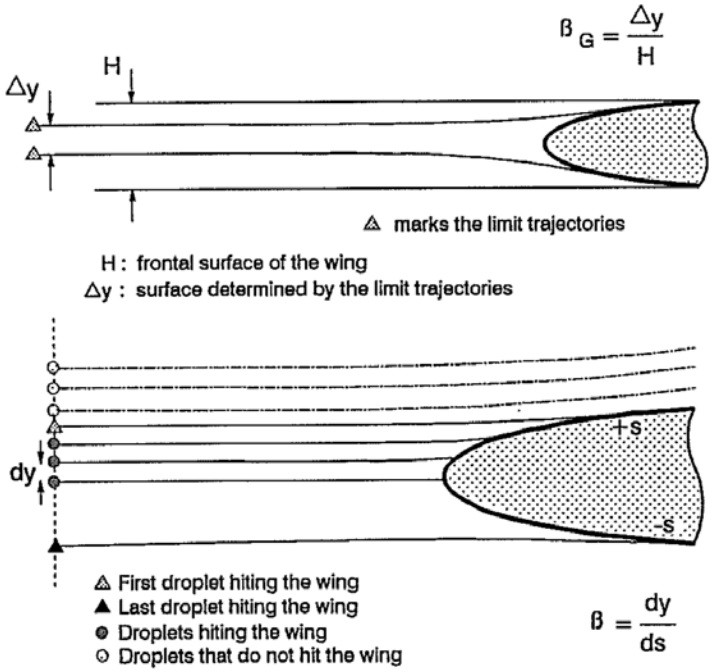

Figure 2.3: Definition of the local and global collection efficiency [2].

Total water mass flow $\dot{m}_e$ impinging to the wing is calculated using the total catch efficiency:

$$\dot{m}_e = \beta_G V_\infty S_F LWC \tag{2.39}$$

where β_G is the global catch efficiency, V_∞ is the free stream velocity, S_F is the airfoil area normal to free stream velocity and LWC is the Liquid Water Content. To find the droplet impact points on the surface, a sweep process is done in (x, y, z) directions. The origin point for the sweep process is the stagnation point. For two-dimensional (x, y) domain, after defining the first droplet impinging the stagnation point, sweep process is performed in y direction and ends when droplets does not impact to the upper and lower wing. For 3D domain, sweep process is done in z direction for each value of y. Impact points are calculated by defining the intersections of each panels with the droplet trajectories, with more details provided in references [2, 49]. Impinging droplet velocity and impinging water mass flow on the surface are used in the icing thermodynamic module for surface mass and energy balance to compute the accreted ice mass.

2.1.2.3 Traditional Messinger model thermodynamics and ice accretion

A mass and energy balance on the impinging surface control volume is performed to calculate the accreted ice on the surface. Solution of the droplet trajectory module (impinging droplet velocity and impinging water mass) along with runback water, the surface heat flux, convective heat transfer coefficient and the other needed terms (such as anti-icing heat fluxes) are used to solve the governing model on the specific surface control volume. The freezing process of the icing simulation has been studied by Messinger [2, 11, 70]. The model is based on the thermodynamic first law of mass and energy conservation on the impinging surface control volume. The mass balance includes mass rate of impinging droplets, mass rate of runback water incoming into the control volume, mass rate of runback water going out of control volume, mass rate of evaporated water or sublimation ice and mass rate of accreted ice. The energy balance includes convective heat, latent heat, evaporation or sublimation heat, and droplet kinetic energy heat (Figure 2.4) [2]. A number of assumptions for the energy balance are:

- Radiation is neglected
- Conduction between surface/water and water/air is neglected
- Runback water temperature is set to surface temperature
- Runback water kinetic energy is neglected

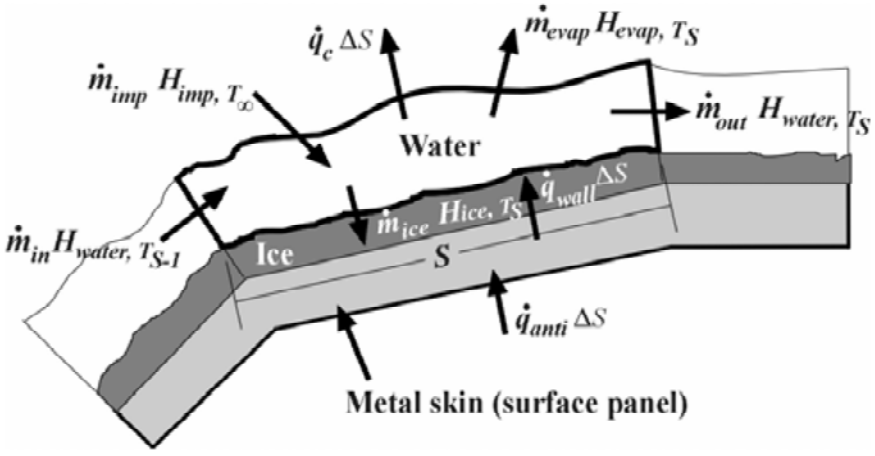

Figure 2.4: Surface control volume for mass and energy balance [5].

Mass and energy balance for the surface control volume are:

$$\dot{m}_{imp} + \dot{m}_{in} = \dot{m}_{ice} + \dot{m}_{evap} + \dot{m}_{out} \tag{2.40}$$

$$\dot{m}_{imp}H_{imp,T_\infty} + \dot{m}_{in}H_{water,T_{s-1}} + \dot{q}_{wall}\Delta s = \dot{m}_{ice}H_{ice,T_s} + \dot{m}_{evap}H_{vapor,T_s} + \dot{m}_{out}H_{water,T_s} + \dot{q}_c\Delta s \tag{2.41}$$

where:

$$H_{imp,T_\infty} = C_{p,water}(T_\infty - 273.15) + \frac{V_\infty^2}{2} \tag{2.42}$$

$$H_{water,T_{s-1}} = C_{p,water}(T_{s-1} - 273.15) \tag{2.43}$$

$$H_{ice,T_s} = C_{p,ice}(T_s - 273.15) - L_f \tag{2.44}$$

$$H_{vapor,T_s} = C_{p,water}(T_s - 273.15) - L_v \tag{2.45}$$

$$\dot{q}_c = h_c\left(T_s - T_e - \frac{V_e^2}{2C_{p,air}}\right) \tag{2.46}$$

$\dot{m}_{imp}$ is the water impinging mass rate, $\dot{m}_{evap}$ is the water evaporation mass rate, $\dot{m}_{ice}$ is the accreted ice mass rate, $\dot{m}_{in}$ and $\dot{m}_{out}$ are the runback water going in and out of the control volume respectively, H is the total enthalpy, $\dot{q}_{wall}$ is the heat rate through the surface, $\dot{q}_c$ is the convective heat rate resulted by airflow and kinetic energy, L_f and L_v are the fusion and evaporation latent heat respectively. The local convective heat coefficient is highly affected by the surface roughness and is calculated using a number of empirical roughness based models [2].

There can be three types of surface depending on the surface temperature: liquid, dry and wet surfaces. Liquid surface means there is no ice accretion on the surface. Dry surface means all the

incoming droplets freezes on impact and there is no water runback. Wet surface means that a portion of the water mass solidifies and the other portion flows as water runback. One way to define the surface type is through a try and false procedure using temperature and freezing fraction terms developed by Messinger [11]. Freezing fraction terms are defined as the portion of the incoming water mass to the control volume that freezes to the total incoming water mass:

$$f = \frac{\dot{m}_{ice}}{\dot{m}_{imp}+\dot{m}_{in}} \tag{2.47}$$

Based on the methodology, for dry rime ice, f = 1, for wet glaze ice, 0 < f < 1, and for liquid surface, f = 0. Defined reference state for thermodynamic balance is water at zero velocity and temperature of 273.15 ºK. Based on the equations (2.40, 2.41, 2.47) there are two unknowns, surface temperature T_s and freezing fraction f. By defining T_s = 273.15 ºK, the freezing fraction is determined from the equations.

- If (f < 0), surface is liquid type with (f = 0) and (T_s > 273.15 ºK), then need to set (f = 0) to recalculate the T_s.
- If (f > 1), surface is dry rime ice type with (f = 1) and (T_s < 273.15 ºK), then need to set (f = 1) to recalculate the T_s.
- If ($0 \leq f \leq 1$) the assumed T_s is correct and surface is wet glaze ice type.

T_s will be calculated in an iterative process until the difference of two successive values is lower than a defined tolerance. By defining the freezing fraction and surface temperature, the other unknowns such as water runback out $\dot{m}_{out}$, ice mass $\dot{m}_{ice}$ and evaporative mass $\dot{m}_{evap}$ are defined from equations.

2.1.3 Grid generation survey

Here a number of well known grid generation tools (general or specific for icing problem) are discussed for the problem of ice structured grid generation and their capabilities and weaknesses are addressed.

2.1.3.1 ICEM CFD

ICEM-CFD is a commercial grid generation tools based on algebraic mesh generation capabilities [42]. It is used for structured, unstructured and hybrid mesh generation in 2&3 dimensional space. The surface preparation is done using the tool CAD capabilities. The blocking is performed by the interactive user interface. The tool has different methods for

surface point's distribution, edge modification, block modification and orientations, etc. ICEM includes replay control setting which allows the user to save/repeat the grid generation process for the similar case studies. The tool includes a number of elliptic smoothing methods such as Steger-Laplace, Steger-Sorenson and Middlecoff-Thomas with limited capabilities such as: local block smoothing, fixed block boundaries, inefficient smoothed grid metrics [73, 74]. The output grid can be generated in a variety of formats for different solvers applications. Figure 2.5 shows a number grids generated using ICEM-CFD commercial tool. The inefficient grid smoothing methods and extensive user work are the main disadvantages of the considered tool.

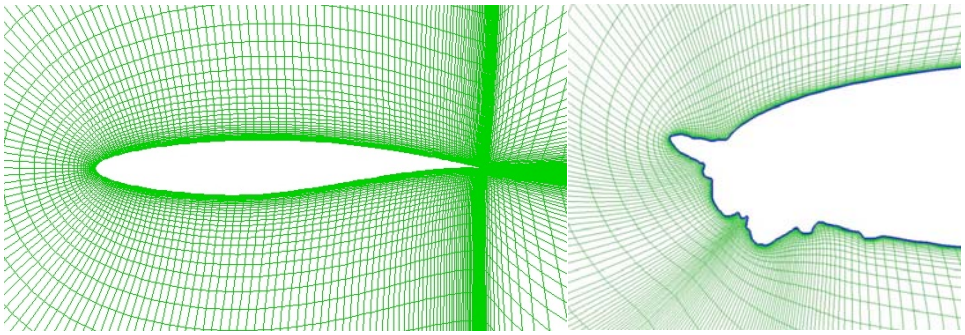

Figure 2.5: Grids generated with ICEM-CFD [31, 39].

2.1.3.2 GRIDGEN

GRIDGEN is a multi-block unstructured/structured elliptic and hyperbolic grid generation commercial tool. The software has been used for different academic and industrial applications [75]. The preliminary Transfinite Interpolation (TFI) grid is generated and used for the elliptic smoothing process. There are different types of elliptic control functions implemented in GRIDGEN, that are similar to the one in ICEM CFD package, such as Thomas-Middlecoeff, Steger-Sorenson, Hilgenstock-White [73]. The surface mesh generation is done using NURBS and PDE elliptic surface generation methods. GRIDGEN is used for various aerospace applications, but has not been used for the ice shapes problems. The hyperbolic grid is based on the front propagation of the grid from the initial surface and is solved via finite difference. The hyperbolic method is based on two main constraints to retain: the trajectory of the propagation and the cell size extension. Figure 2.6 shows few grid cases generated using GRIDGEN tool. The grid shock problem in concave area can be seen in Figure 2.6.

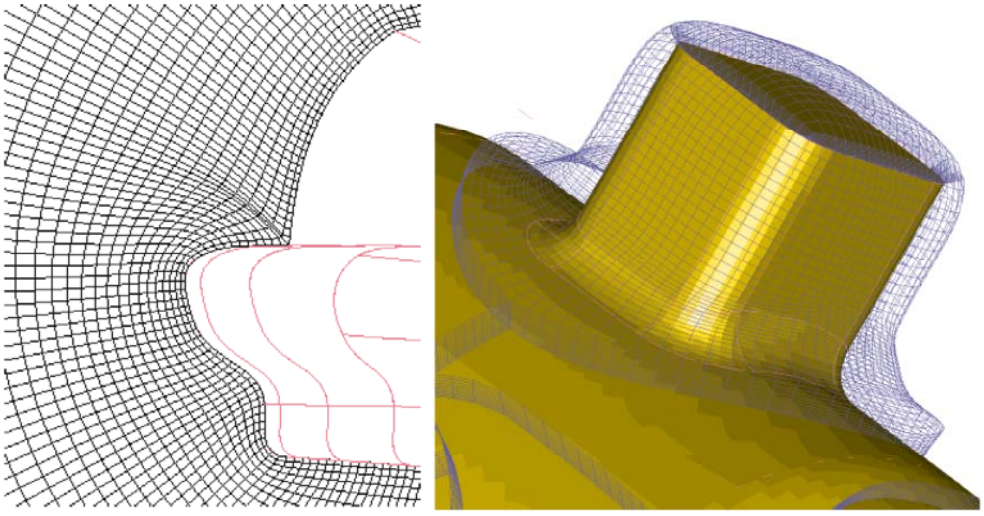

Figure 2.6: Mesh generated using GRIDGEN, elliptic (left), hyperbolic (right) [75].

2.1.3.3 SmaggIce (Specific for icing)

There are a number of proposed approaches for ice grid quality improvements using different blocking topologies and domain decomposition methods that are applied in grid generation software developed specifically for ice effect analysis. SmaggIce software has been developed by NASA for this purpose [76, 77]. Preparation of the geometry surface needs to be properly done because of complexity of the ice shapes with horns and sharp edges. The blocking topology is defined as either single-block or multi-block. Single-block topology results in grid clustering issues in concave region that reduces the accuracy of the flow solution and increases the grid induced errors. Multi-block topology results in extension of the clustering along the block boundaries caused by grid cluster on the solid walls (Figure 2.4). These high aspect ratio cells are extended into the domain along the block boundaries, producing considerable errors in the flow solution. For the small scale roughness on the ice surface, both of the single and multi-block approaches have the difficulty to contain high quality grid on the surface and propagation of the surface discontinuities to the domains.

Shih [6, 25] developed a new blocking strategy implemented in SmaggIce to overcome these difficulties. The strategy for the single-block includes a generation of domain grids which contains two sections: fine grids close to the airfoil and coarse grids in the farfield. In fine grid section, a grid line away from iced surface is chosen and all the cells in between (between the chosen line and iced surface) are regenerated and smoothed. The inner block close to the surface can contain many numbers of sub-blocks to increase the quality of the grid close to the iced surface (Figure 2.7). At the end of grid generation process, all blocks are combined as a single

block grid. The approach efficiently reduces the grid clustering in the domain but it efforts extensive manipulation such as line decision, sub-blocking and smoothing. Automation of the process is one of the main difficulties of the proposed approach.

Developed strategy for the multi-block grid generation includes a thick wrap-around layer that covers the iced surface and removes the extension of the wall grids clustering along the block boundary into the domain (Figures 2.7 and 2.8). To solve the difficulty of surface small roughness propagation into the domain, a transition surface is used to smooth the grid in between (Figures 2.9). This strategy also needs to be applied manually such as creating the thick wrap-round layer, blocking and defining transition layer. The difficulty of automation is one of the main problems that reduces the flexibility of the approach for multi-time step icing RANS based methodologies.

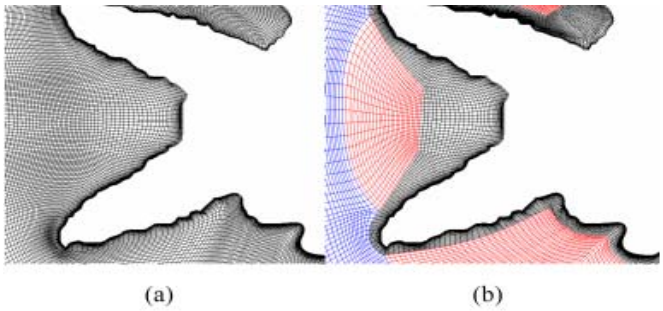

Figure 2.7: Grid generated for a glaze ice shape: (a) Single-block. (b) Multi-block [77].

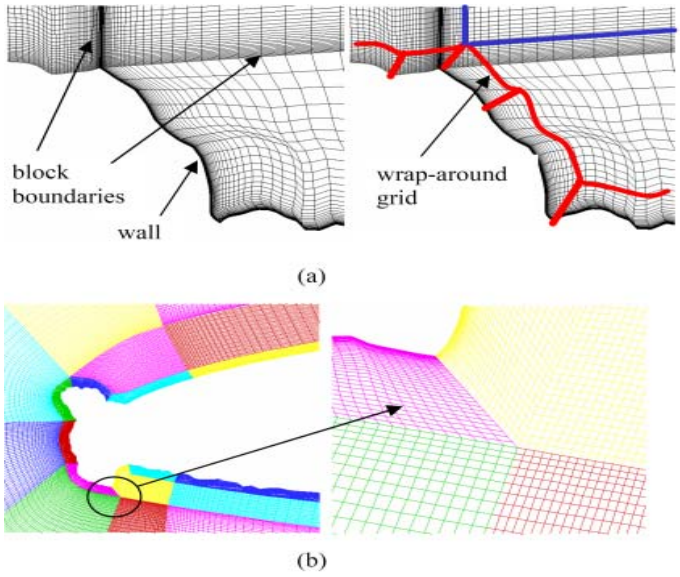

Figure 2.8: Grid generated and modified for glaze ice shape: (a) Grid line cluster next to the wall propagates into the domain along the block boundaries, (b) Effect of wrap-around to remove the clustering via the multi-block decomposition [77].

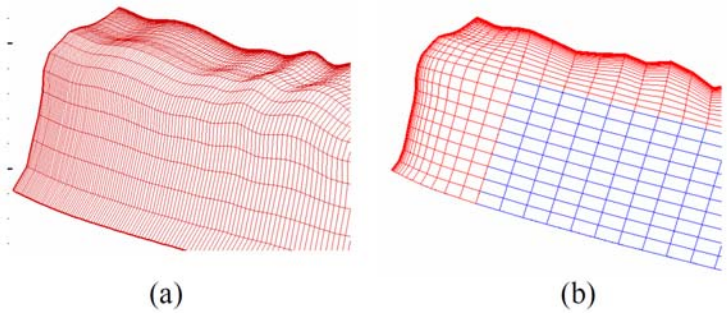

Figure 2.9: Grid generated and modified for a glaze ice shape: (a) Surface oscillation propagated into domain, (b) Transition layer removed the surface oscillation propagation [77].

2.1.3.4 TurboGrid

Turbo-Grid [78] is an interactive tool performing single and multi-block structured algebraic grid generation with surface and simple field Laplace/elliptic smoothing. Use of control points, stretching, refining, boundary manipulation on the workstation monitor gives the flexibility to the user to control the algebraic grid quality. However, when applied on complex geometries such as experimental glaze ice shapes, the tool suffers from heavy dependence on user inputs to improve the grid quality. Note the PDE smoothing capability of the tool is very primitive. Figure 2.10 shows a grid generated and modified for an ice case study using TurboGrid.

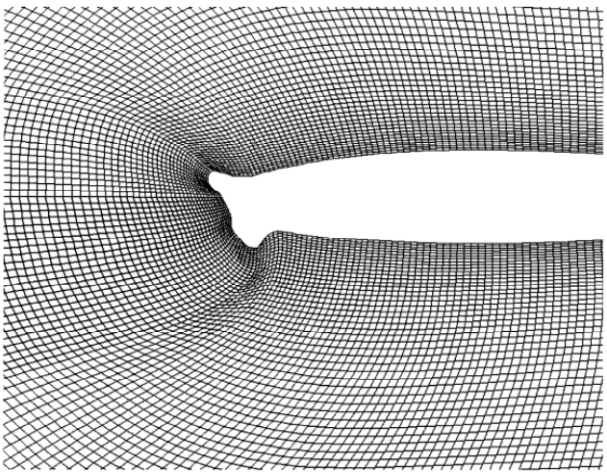

Figure 2.10: Grid generated and modified for a glaze ice shape with Turbo-Grid [78].

2.1.3.5 ICEG2D (Specific for icing)

Conformal mapping approaches such as parabolic mapping methods are one of the grid generation classes discussed for the problem of ice grid generation. The parabolic structured and semi-structured grid generation approach developed by Thompson and Soni shows improved flexibility for two-dimensional and three-dimensional ice shapes grid generation [58, 79-81]. The

structured parabolic grid generation code has been used in LEWICE-NS integrated with NPARC solver to perform RANS based multi-time steps icing simulation ICEG2D [12]. Parabolic method starts by algebraically generation of two reference layers that are locally orthogonal. These layers have been smoothed using Poisson equation with control functions. After smoothing the reference layers, the last 2 grid layers are used to generate the next grid layer algebraically and then smoothed with Poisson equation. This procedure is repeated until the outer boundaries are reached. A 2D structured grid generated for glaze ice shapes using the proposed parabolic approach is shown in Figure 2.11. Note that the grid shock still poses challenges.

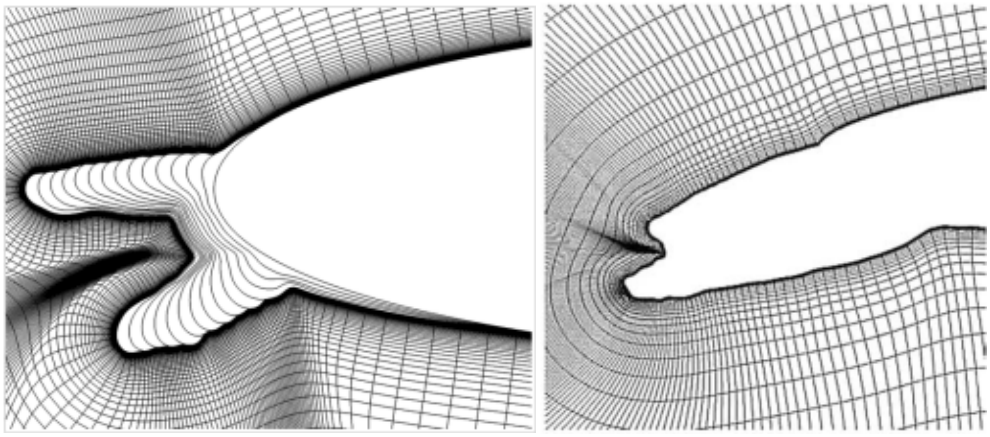

Figure 2.11: Structured grid generated for two glaze ice shapes [12, 58].

Semi-structured grids have been examined using the same parabolic approach including additional curvature effect and line modification functions. Semi-structured meshes are types of hybrid grids that combines structured and unstructured concepts and can be used as an alternative approach for the problem of ice grid generation [80]. Hybrid grids are more flexible than structured grids for complex geometries. They also have higher resolution and lower elements compared to unstructured grids and, as unstructured grids, are based on an explicit connectivity table. The performance of the approach in flow simulation around the iced airfoil is tested by Thompson and Soni [79]. Semi-structured grids generate quad/hexahedral elements. The algorithm is based on parabolic marching scheme and generation of grids from initial data surface. It includes three steps: algebraic generation of locally orthogonal reference mesh, smoothing the reference mesh using Poisson equation, and finally deletion and insertion algorithm based on the reference grid quality. The resulted mesh is highly attractive for viscous flow calculation and near body region resolution. The parabolic structured and semi-structured methodologies are given here [12, 79]. Assuming the transformation is the form of:

$$\xi = \xi(x,y,z)\ ,\ \eta = \eta(x,y,z)\ ,\ \zeta = \zeta(x,y,z) \tag{2.48}$$

the inverse transformation is:

$$x=x(\xi,\eta,\zeta)\ ,\ y=y(\xi,\eta,\zeta)\ ,\ z=z(\xi,\eta,\zeta) \tag{2.49}$$

The initial data surface is assumed to be ζ=constant surface and ζ is chosen as marching direction.

The approach is as follows: First step: a structured reference mesh is algebraically generated to be locally orthogonal. This reference mesh includes three layers (subscript 0, 1, 2): one initial data surface and two grid layers. The first surface mesh for layer k is:

$$r^k_{i,j,1} = r^k_{i,j,0} + \delta^k_{i,j}.\, n^k_{i,j,0} \tag{2.50}$$

where $r^k_{i,j,0}$ is the position vector of point (i,j) on the initial data surface, $r^k_{i,j,1}$ is the position vector of point (i,j) on the first surface of reference mesh, $\delta_{i,j}$ is the specified distance distribution, and $n_{i,j,0}$ is the unit surface normal. The second surface of reference mesh for layer k is:

$$r^k_{i,j,2} = r^k_{i,j,1} + \delta^{k+1}_{i,j}.\, n^k_{i,j,1} \tag{2.51}$$

Second step: the standard Poisson equation is used to smooth the generated reference mesh layers. The orthogonality, spacing and curvature control are included in control functions to improve the quality of grids. The smoothing is only applied on the points of the first surface of the reference mesh, and the second surface is updated after completing the desired number of iterations for smoothing of the first surface. Based on the orthogonality assumption, g_{13}, g_{12}, and g_{23} should be zero, so that the Poisson equation reduces to:

$$\frac{g_{22}g_{33}}{g^2}\left(r_{\xi\xi} + \Phi r_\xi\right) + \frac{g_{11}g_{33}}{g^2}\left(r_{\eta\eta} + \Psi r_\eta\right) + \frac{g_{11}g_{22}}{g^2}\left(r_{\zeta\zeta} + \Theta r_\zeta\right) = 0 \tag{2.52}$$

$$g_{11} = x_\xi^2 + y_\xi^2 + z_\xi^2\ ,\ g_{22} = x_\eta^2 + y_\eta^2 + z_\eta^2\ ,\ g_{33} = x_\zeta^2 + y_\zeta^2 + z_\zeta^2 \tag{2.53}$$

$$g = \begin{vmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{vmatrix} \tag{2.54}$$

where g_{11}, g_{22}, g_{33} are covariant metrics components; x_ξ, y_ξ, z_ξ are the position derivatives to ξ-direction; x_η, y_η, z_η are the position derivatives to η-direction; x_ζ, y_ζ, z_ζ are the position derivatives to ζ-direction.

The form of control functions with the curvature effects are:

$$\Phi = -\frac{\partial}{\partial\xi}\ln\left(\sqrt{\frac{g_{11}}{g_{22}g_{33}}}\right) \tag{2.55}$$

$$\Psi = -\frac{\partial}{\partial \eta} \ln\left(\sqrt{\frac{g_{22}}{g_{11}g_{33}}}\right) \tag{2.56}$$

$$\Theta = -\frac{\partial}{\partial \zeta} \ln\left(\sqrt{\frac{g_{33}}{g_{11}g_{22}}}\right) \tag{2.57}$$

The control functions are computed on all layers. The initial data surface is used to define the value of Φ, Ψ, and all three surfaces in the layer are used to define the value of Θ. For highly non-convex regions, it has been found that dissipation should be added to the Poisson smoothing equation [81]. In the domains where g11>>g33, the derivatives of ζ are dominated in the Poisson equation, so that in non-convex domains this would results in lower effects of smoothing in the ζ constant lines which lead to the grid lines crossing problem. This problem is solved using a dissipation term to modify the Poisson equation first term as follows:

$$\frac{g_{22}g_{33}}{g^2}\left((1+\nu_\xi)r_{\xi\xi} + \Phi r_\xi\right) \tag{2.58}$$

where:

$$\nu_\xi = \sqrt{\frac{\max\left(g_{11}g_{33}\right)}{g_{33}}} \times f(\theta_\xi) \tag{2.59}$$

$$f\left(\theta_\xi\right) = \begin{cases} 1 & 0 \le \theta_\xi < \frac{\pi}{2} \\ (sin\theta_\xi)^\alpha & \frac{\pi}{2} \le \theta_\xi < \pi \\ 0 & \pi \le \theta_\xi \end{cases} \tag{2.60}$$

The term θ_ξ is the angle between $r_{i+1,j} - r_{i,j}$ and $r_{i-1,j} - r_{i,j}$. The value of $0 < \alpha < 1$ is defined by user to control the smoothing in the concave domains.

The process of structured parabolic method is thus completed. Now to move forward with semi-structured grids, the following additional step is defined: Third step: a geometry based line deletion/insertion algorithm is applied on the smoothed reference mesh. Two criteria are chosen for deletion of a line such as ξ line. The first condition is the slenderness of the cell and if the condition is true, the ξ=constant line is deleted:

$$\frac{1}{j_{max}} \sum_{j=1}^{j_{max}} \left(\sqrt{\frac{g_{33}}{g_{11}}}\right)_j > \alpha_{avg} \tag{2.61}$$

This translates as a cell on the ξ=constant line where the average value of the ratio of ξ arclength to ζ arclength should be lower than the specified value of α_{avg}. Otherwise, the ξ constant line will be deleted. The j_{max} is the number of points on the ξ=constant line. The second condition is based on the cell maximum slenderness on the ξ lines and defined as the ratio of ξ arclength to ζ

arclength which needs to be lower than the specified value of α_{max}. Otherwise, the ξ constant line is deleted.

$$max\left(\sqrt{\frac{g_{33}}{g_{11}}}\right)_j > \alpha_{max} \tag{2.62}$$

Note that the user specified values need to be chosen with the condition ($\alpha_{max} > \alpha_{ang}$).

For the insertion algorithm, the criteria are based on the cell face divergence:

$$\Delta = \frac{1}{\sqrt{g_{33}}}\frac{\partial}{\partial \zeta}(\sqrt{g_{11}}) \tag{2.63}$$

$$\frac{1}{j_{max}}\sum_{j=1}^{j_{max}}(\Delta)_j > \beta_{avg} \tag{2.64}$$

where Δ is the cell face divergence. If the average divergence condition is higher than the user specified value of β_{avg}, a ξ line will be added to the surface of ξ=constant and ζ=constant. The second condition is specified as below:

$$max(\Delta)_j > \beta_{max} \tag{2.65}$$

If the condition is true, a ξ line will be added to the surface of ξ=constant and ζ= constant. Note that user specified parameters β_{max} and β_{avg} should follow the condition ($\beta_{max} > \beta_{ang}$).

The deletion/insertion algorithm in two dimensional domain is shown in Figure 2.12. Points 1, 2, and 3 are on the initial data surface for the layer k and points 4, 5, and 6 are generated algebraically and smoothed as discussed above. For the next layer, point 5 is not chosen for the generation of the next layer k+1, based on the criteria of the deletion/insertion algorithm. Only points 4 and 6 are used as initial data surface for generation of the next layer k+1 which results in points 7 and 8.

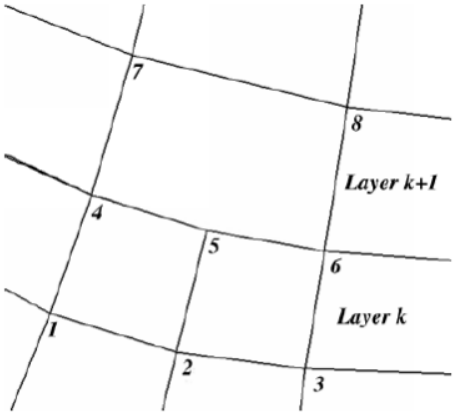

Figure 2.12: Deletion/insertion algorithm in two dimensional grids [79].

The cell generated by deletion/insertion condition can have 4 to 6 edges in their faces and their volume can have 6 to 17 faces. Decision of the cell aspect ratio as the criteria for deletion/insertion algorithm gives the flexibility to control the cell aspect ratio close to the body suitable for viscous dominated region. The main disadvantage of the semi-structured approach is that there are not many flow solvers capable of using this type of grids. Semi-structured generated grids for different 2D and 3D geometries are shown in Figure 2.13. Also the generated 2D grids for two glaze ice shapes are shown in Figure 2.14. The grid shock is well addressed by this approach.

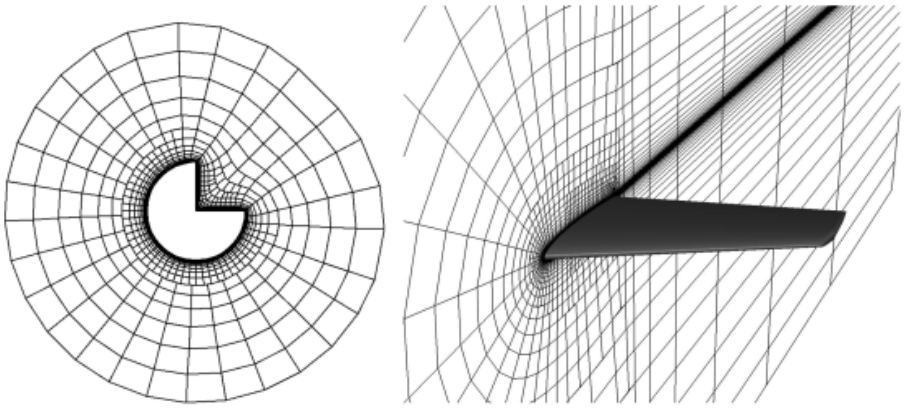

Figure 2.13: Semi-structured grids for arbitrary 2D and 3D geometries [79, 80].

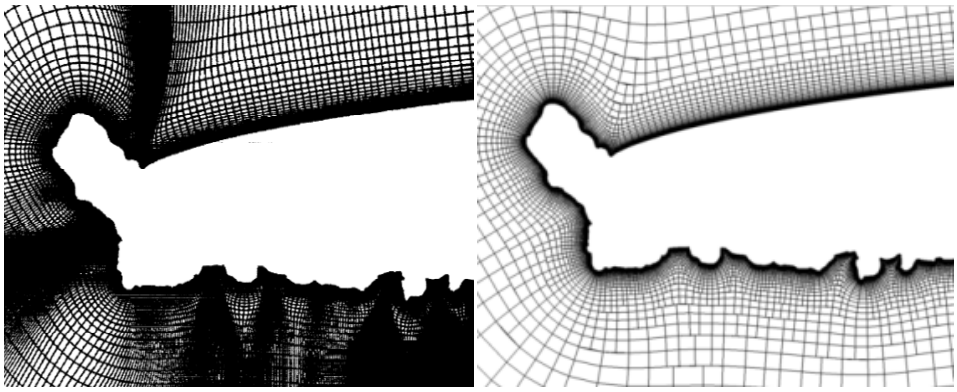

Figure 2.14: Structured grid (left) and Semi-structured grid (right), 2D glaze ice shapes [79, 80].

2.1.3.6 EGRID

The multi-block structured elliptic grid generation EGRID is developed at Bombardier Aerospace [56]. The code has an advanced domain decomposition algorithm to solve the blocking topology problem. The tool is capable of algebraic grid generation, grid manipulation and improvements. It also includes an elliptic smoothing method with stretching, orthogonality and curvature control functions.

Considering ξ^i is the curvilinear coordinates and (n) is the direction normal to the ξ^i = constant curve, curvilinear coordinates must satisfy the following equation to remove the Laplace equation tendency to alter the mesh spacing and curvature in concave or convex regions:

$$\frac{\partial^2 \xi^i}{\partial n^2} = 0 \; ; \; \text{i} = 1, 2, 3 \tag{2.66}$$

For uniform spacing grids, the expressions for normal derivative and derivative for metric tensor are:

$$\frac{\partial \xi^i}{\partial n} = \frac{1}{\sqrt{g^{ii}}} g^{ij} \frac{\partial \xi^i}{\partial \xi^j} \qquad , \qquad \frac{\partial g^{ii}}{\partial \xi^i} = -2\Gamma^i_{jk} g^{ik} \tag{2.67}$$

The term Γ^i_{jk} are the Christoffel symbols [15]. The equation simplifies to:

$$\frac{\partial^2 \xi^i}{\partial n^2} = -\Gamma^i_{jk} g^{ii} g^{ik} / g^{ii} = 0; \quad \text{i} = 1, 2, 3 \tag{2.68}$$

This equation is similar to the curvilinear Laplace equation ($\nabla^2 \xi^i = -\Gamma^i_{jk} g^{ik} = 0$). Considering the equivalent Poisson equation form as:

$$\nabla^2 \xi^i = C^i \; ; \; \text{i} = 1, 2, 3 \tag{2.69}$$

where C^i is the curvature control function. By adding the equation (2.30) to the Poisson equation, and using the simplification addressed in literature [56], the curvature control function can be defined as:

$$C^i = -(K_1^{(i)} + K_2^{(i)}) \sqrt{g^{ii}} \tag{2.70}$$

$$K_1^{(i)} + K_2^{(i)} = \frac{1}{g (g^{ii})^{\frac{3}{2}}} (\Gamma^i_{jj} g_{kk} + \Gamma^i_{jk} g_{jk} + \Gamma^i_{kk} g_{jj}) \tag{2.71}$$

where the $K_1^{(i)}$ and $K_2^{(i)}$ are the local principal curvatures of the ξ^i constant surface.

The Poisson equation with curvature control function for uniform mesh simplifies to:

$$\nabla^2 \xi^i = -(K_1^{(i)} + K_2^{(i)}) \sqrt{g^{ii}} \quad ; \quad \text{i} = 1, 2, 3 \tag{2.72}$$

Note that the new Poisson equation system reduces to Laplace equation by removing the curvature function.

To extend the curvature function to non-uniform spacing mesh, an elliptic equation with function controlling the spacing (S^i) is used:

$$\nabla^2 s^i = 0 \; ; \; \text{i} = 1, 2, 3 \tag{2.73}$$

To remove the undesirable curvature effects, S^i must satisfy the equation below, similar to equation (2.66) for ξ^i:

$$\frac{\partial^2 s^i}{\partial n^2} = 0 \; ; \; \text{i} = 1, 2, 3 \tag{2.74}$$

An equivalent Poisson system for function S^i is:

$$\nabla^2 s^i = C^i \; ; \; i = 1, 2, 3 \tag{2.75}$$

$$C^i = -(\bar{K}_1^{(i)} + \bar{K}_2^{(i)})\sqrt{\bar{g}^{ii}} \tag{2.76}$$

where $\bar{K}_1^{(i)}$, $\bar{K}_2^{(i)}$, and $\bar{g}^{ii}$ are functions of S^i. By solving the Laplacian of S^i as a function of curvilinear coordinates ξ^i, we have:

$$\nabla^2 s^i = g^{jk} \frac{\partial^2 s^i}{\partial \xi^j \partial \xi^k} + \nabla^2 \xi^l \frac{\partial s^i}{\partial \xi^l} = C^i \tag{2.77}$$

The term $\nabla^2 \xi^l$ can be written as:

$$\nabla^2 \xi^l = C^i \frac{\partial \xi^l}{\partial s^i} - g^{jk} \frac{\partial \xi^l}{\partial s^i} \frac{\partial^2 s^i}{\partial \xi^j \partial \xi^k} \tag{2.78}$$

In the right-hand side of above equation, the first term is the general form of curvature control function for non-uniform spacing mesh, and the second term is the function to control the mesh spacing. For uniform mesh this equation reduces to equation (2.72).

In general, for generating non-uniform spacing grids with curvature effects, the equation system is written as:

$$\nabla^2 r = g^{jk} \frac{\partial^2 r}{\partial \xi^j \partial \xi^k} + \nabla^2 \xi^l \frac{\partial r}{\partial \xi^l} = 0 \tag{2.79}$$

where r = (x,y,z) and $\nabla^2 \xi^l$ is defined by equation (2.78).

The presented Poisson equation with curvature function has been tested for different case studies. The effect of curvature control is shown in Figure 2.15.

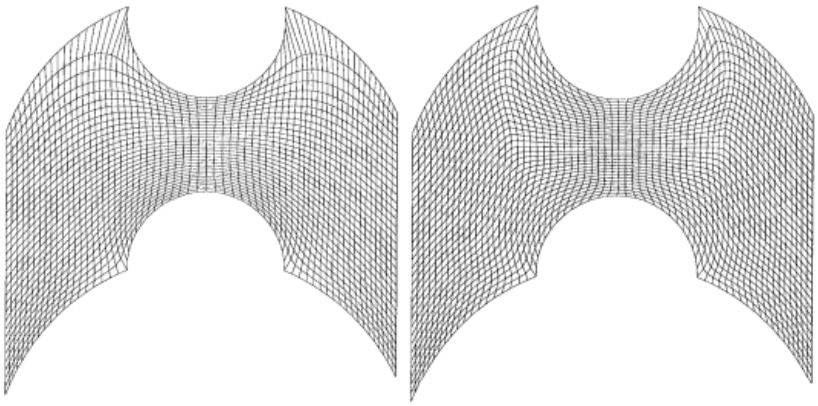

Figure 2.15: Grid generated without curvature effect (left), and with curvature effect (right) [56].

2.1.3.7 GRAPE2D

GRAPE a multi-block mesh generation/smoothing based on Poisson elliptic method. The control functions incorporated are developed by Sorenson [55, 74, 82, 83]. Figure 16 shows a

number of grid generated using GRAPE code [55]. In the Sorenson approach (RLS), P_{RLS} and Q_{RLS} are defined as:

$$P_{RLS}(\xi,\eta) = p(\xi)\cdot e^{-\alpha\eta} + r(\xi)\cdot e^{-c(\eta_{max}-\eta)} \tag{2.80}$$

$$Q_{RLS}(\xi,\eta) = q(\xi)\cdot e^{-b\eta} + s(\xi)\cdot e^{-d(\eta_{max}-\eta)} \tag{2.81}$$

where α, b, c and d are positive constants defining the propagation of sources on boundaries through domain.

Two main control functions, grid spacing and grid orthogonality have been considered. The minimum spacing close to wall along ξ = constant is defined by the user as Δs or ds to control the grid spacing distribution. Also the grid angle θ of layer η = 0 (inner boundary) has been defined by user to control the orthogonality of the grid on the body and the propagation through domain. For spacing control (ds), on the layer η = 0 (inner boundary), the equations are:

$$ds = [(dx)^2 + (dy)^2]^{0.5} \tag{2.82}$$

Substitution the partial differentiations for dx and dy:

$$ds = \left[\left(x_\xi d\xi + x_\eta d\eta\right)^2 + \left(y_\xi d\xi + y_\eta d\eta\right)^2\right]^{0.5} \tag{2.83}$$

As $d\xi = 0$ (ξ = constant) and $d\eta = 1$ for the considered interval, the equation simplifies to:

$$ds = \left[\left(x_\eta\right)^2 + \left(y_\eta\right)^2\right]^{0.5} \tag{2.84}$$

For orthogonality control (θ), the angle between the first layer η = 0 (inner boundary) and ξ has been defined as follows:

$$[\nabla\xi\cdot\nabla\eta] = [|\nabla\xi|\,|\nabla\eta|\cos\theta] \tag{2.85}$$

which results in:

$$\left[\xi_x\eta_x + \xi_y\eta_y\right] = \left[\left(\xi_x{}^2 + \xi_y{}^2\right)^{0.5}\left(\eta_x{}^2 + \eta_y{}^2\right)^{0.5}\cos\theta\right] \tag{2.86}$$

Using the transformation correlation (4,5), the equation simplify to:

$$\left[-y_\eta y_\xi - x_\eta x_\xi\right] = \left[\left(y_\eta{}^2 + x_\eta{}^2\right)^{0.5}\left(y_\xi{}^2 + x_\xi{}^2\right)^{0.5}\cos\theta\right] \tag{2.87}$$

Importing the equation of ds; above correlation for the layer η=0 (inner boundary) will reduce to:

$$x_\eta = \left[\frac{ds(-x_\xi\cos\theta - y_\xi\sin\theta)}{\left(x_\xi{}^2 + y_\xi{}^2\right)^{0.5}}\right] \tag{2.88}$$

$$y_\eta = \left[\frac{ds(-y_\xi\cos\theta - x_\xi\sin\theta)}{\left(x_\xi{}^2 + y_\xi{}^2\right)^{0.5}}\right] \tag{2.89}$$

The same procedure can be applied for the source term on the last layer $\eta = \eta_{max}$ (outer boundary).

Back to the P_{RLS}, Q_{RLS} equations, based on the assumptions (using damping function controlled by the parameters α, b, c, d), the effect of the terms r and s should be negligible on the layer η=0 (inner boundary), and similarly the negligible effect of terms p and q on the layer $\eta = \eta_{max}$ (outer boundary). This assumption results in:

On the inner boundary (η = 0):

$$P_{RLS}(\xi,0) = p(\xi) \text{ and } Q(\xi,0) = q(\xi) \tag{2.90}$$

Using the above correlations and Poisson transformed equation, results in:

$$p(\xi) = \left[\frac{y_\eta\left(-\left(\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}\right)/J^2\right) - x_\eta\left(-\left(\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta}\right)/J^2\right)}{J}\right] \tag{2.91}$$

$$q(\xi) = \left[\frac{-y_\xi\left(-\left(\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}\right)/J^2\right) + x_\xi\left(-\left(\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta}\right)/J^2\right)}{J}\right] \tag{2.92}$$

Similarly, on the outer boundary ($\eta = \eta_{max}$), the terms r and s can be computed by the correlation $P_{RLS}(\xi,\eta_{max}) = r(\xi)$, $Q_{RLS}(\xi,\eta_{max}) = s(\xi)$ and Poisson transformed equation.

Using the computed derivatives, the terms p, q, r, s are determined and finally sources P_{RLS} and Q_{RLS} are computed through the domain. The iterative process proposed by Sorenson [82] starts by using an algebraic generated grid as the initial solution and the user specified spacing and stretching parameters. The initial values for p, q, r, s are assumed to be zero and input values for Δs and θ are defined. All derivatives of the boundary grid points for the first iteration are computed. The solution of the first iteration is used in the next iteration to solve $x_{\eta\eta}$, $y_{\eta\eta}$ and other derivatives on the boundaries. The values of p, q, r, s for the boundary points are then determined. Now P_{RLS} and Q_{RLS} can be computed for all the interior grid points using Sorenson control function equations. The terms p, q, r, s of the defined control functions can be under-relaxed to increase the numerical stability. The elliptic Poisson PDE equation system can be solved using any solution approach (implicit or explicit such as point and line SOR, Point Jacobi and ADI, with additional FMG method for speeding up the computation). This iterative process is continued to satisfy a convergence criterion.

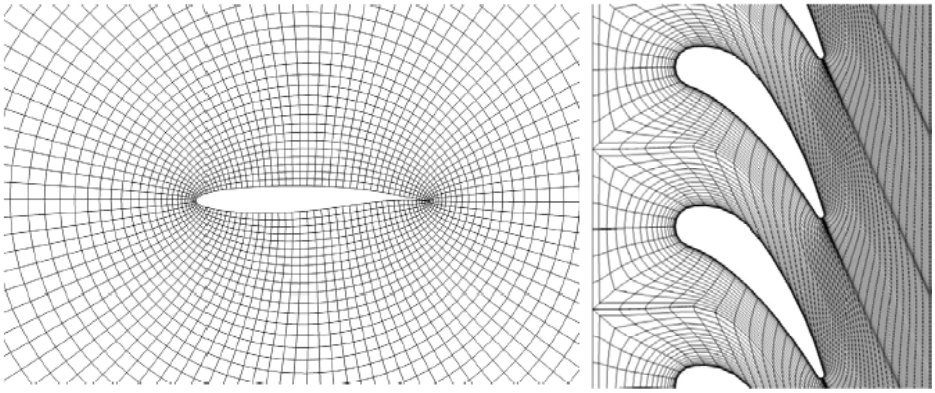

Figure 2.16: GRAPE generated mesh [55].

2.1.3.8 ENGRID

ENGRID, developed at NLR, is a multi-block structured grid generation based on algebraic and elliptic grid methods. The main elliptic method implemented in ENGRID is the Spekreije approach including spacing, orthogonality and curvature grid control. The equations are discretized using Finite Difference and Finite Volume methods. An efficient black-box Multi-Grid solver is used to reduce the computation time [22, 84]. The elliptic method control functions are computed using the parametric space. The mapping domains are shown in Figure 2.17. The parametric domain is defined by nonlinear algebraic transformations. Using the preliminary defined algebraic and parametric domains, the source terms are computed once for the first iteration and held fixed throughout the entire grid computation.

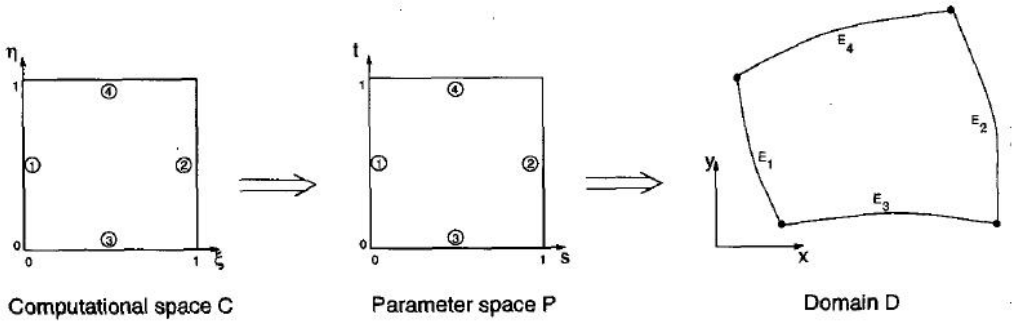

Figure 2.17: Composite mapping [84].

The Poisson with control function equations are as follows:

$$a^{11}X_{\xi\xi} + 2a^{12}X_{\xi\eta} + a^{22}X_{\eta\eta} + P_{SPS}X_{\xi} + Q_{SPS}X_{\eta} = 0 \tag{2.93}$$

$$P_{SPS}(\xi,\eta) = a^{11}p_{11}^{1} + 2a^{12}p_{12}^{1} + a^{22}p_{22}^{1} \tag{2.94}$$

$$Q_{SPS}(\xi,\eta) = a^{11}p_{11}^{2} + 2a^{12}p_{12}^{2} + a^{22}p_{22}^{2} \tag{2.95}$$

$$p_{11} = -T^{-1}\begin{pmatrix} s_{\xi\xi} \\ t_{\xi\xi} \end{pmatrix},\ p_{12} = -T^{-1}\begin{pmatrix} s_{\xi\eta} \\ t_{\xi\eta} \end{pmatrix},\ p_{22} = -T^{-1}\begin{pmatrix} s_{\eta\eta} \\ t_{\eta\eta} \end{pmatrix},\ T = \begin{pmatrix} s_{\xi} & s_{\eta} \\ t_{\xi} & t_{\eta} \end{pmatrix} \quad (2.96)$$

$$a^{11} = \left(x_{\eta}{}^2 + y_{\eta}{}^2\right)/J^2 \qquad a^{22} = \left(x_{\xi}{}^2 + y_{\xi}{}^2\right)/J^2 \qquad a^{12} = \left(x_{\xi}x_{\eta} + y_{\xi}y_{\eta}\right)/J^2 \quad (2.97)$$

where a^{11}, a^{12}, a^{22} are the covariant metrics components and p_{11}, p_{12}, p_{22} are the control function terms.

To incorporate the orthogonality, the elliptic equation below is solved using FVM and the defined boundary conditions from the orthogonality criteria on physical space (Neumann boundary conditions $(\partial s/\partial n = 0)$ for edges s=0 and s=1 and $(\partial t/\partial n = 0)$ for edges t=0 and t=1.

$$\left(Ja^{11}s_{\xi} + Ja^{12}s_{\eta}\right)_{\xi} + \left(Ja^{12}s_{\xi} + Ja^{22}s_{\eta}\right)_{\eta} = 0 \quad (2.98)$$

$$\left(Ja^{11}t_{\xi} + Ja^{12}t_{\eta}\right)_{\xi} + \left(Ja^{12}t_{\xi} + Ja^{22}t_{\eta}\right)_{\eta} = 0 \quad (2.99)$$

Once s and t are computed on the boundaries, they are used to regenerate the parametric grid domain by Hermit interpolation:

$$s = s_{E_1}(\xi)H_0(t) + s_{E_3}(\xi)H_1(t) \quad (2.100)$$

$$t = s_{E_2}(\eta)H_0(t) + s_{E_4}(\eta)H_1(t) \quad (2.101)$$

$$H_0(s) = (1 + 2s)(1 - s)^2\ ,\ H_1(s) = (3 - 2s)s^2 \quad (2.102)$$

$$H_0(t) = (1 + 2t)(1 - t)^2\ ,\ H_1(t) = (3 - 2t)t^2 \quad (2.103)$$

Considering the new generated parametric space (s, t), terms P and Q are computed again for the first iteration and applied through the grid computation to impose the condition of orthogonality in the physical space. ENGRID has been used for different academic and industrial aerospace applications, but not for complex ice configurations. Figure 2.18 shows few generated grids by ENGRID meshing tool, which shows a poor resolution in highly concave regions.

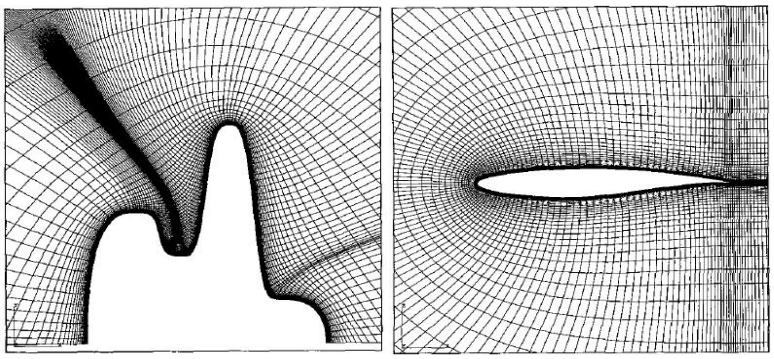

Figure 2.18: ENGRID generated mesh [22].

2.2 Methodology

Here we present the detailed methodology of the RANS based icing simulation developments performed in this thesis.

2.2.1 Ice grid generation

Here we propose algorithms for the ice grid problem, in particular the treatment of concave/convex regions to reduce the grid shock phenomena as well as choice of control functions to achieve appropriate grid metrics in the field. The solutions presented are moreover constrained by a requirement to be fully automated so they can eventually be used within a multi-steps icing calculation procedure. The 1D curvilinear curvature based algorithm is presented, followed by the field generation algorithms (parabolic, elliptic). A detailed account of the various control functions is made and a blended approach is proposed. The numerical solution of the governing Poisson equations is also presented.

2.2.1.1 Surface adaptive curvature based grid point distribution algorithm

Here a novel 1D elliptic geometry curvature based point distribution algorithm is described [41]. Considering the 2D transformed Poisson equation, a 1D elliptic transformed equation with curvature based source terms is developed to satisfy the curvature based point distribution on the iced body:

$$r_{\xi\xi} = S \tag{2.104}$$

where r is the geometry point distribution arclength (normalized length, $0 \leq r \leq 1$), and S is the source term. The normalized length formulation is as follow:

$$r_i = (r_{i-1} + \Delta r)/r_n \tag{2.105}$$

$$\Delta r = \sqrt{(x_i + x_{i-1})^2 + (y_i + y_{i-1})^2} \tag{2.106}$$

A number of source functions such as Sign, Sine, Cosine, Spline, damping Cosine, damping Atangent, damping Spline, mixed functions, etc. have been analyzed. Periodic source terms functions are chosen to achieve clustering at and near high curvature maxima, while unclustering between 2 high curvature maxima. Here a simple efficient periodic function is heuristically selected as the source term to add to the right hand side of the elliptic PDE:

$$S = A \cdot (r + \sin(2\pi r)) \tag{2.107}$$

where A is the amplitude constant parameter defined by the user. This parameter adjusts the global intensity of the surface grid points on the curved domains and can change the stability of

the numerical elliptic scheme if chosen very large; it is defined as ($0 \leq A \leq 0.001$) for the case studies presented here.

The step by step procedure of the proposed 1D elliptic curvature based PDE computation is as follows. First, the surface curvature is computed by a finite number of points uniformly distributed on the surface. The curvature value is defined as:

$$\mathrm{Curv_i} = -1/\mathrm{Radius_i} \tag{2.108}$$

where $\mathrm{Radius_i}$ is the computed radius of the circle mapped on 3 points (i-1, i, and i+1). For each point (and its arclength) the curvature is defined concave (negative) or convex (positive). Then, a number of points (k=2 to n-1) is selected by the user, depending on the curvature selection criterion, to apply the curvature source term S in a piecewise fashion. Note that k=1 and k=n are the first and last point on the 1D geometry. Second, the source term S is computed for the selected geometry points (k=1 to n):

$$S1_i = A \cdot \mathrm{Sin}\left(\frac{r_i - r_k}{r_k - r_{k-1}} \cdot 2\pi\right) \quad \text{for } (k = 2 \text{ to } n\text{-}1) \text{ and if } (r_{k-1} \leq r_i \leq r_k)$$

$$S2_i = A \cdot (\mathrm{Curv_i}/\textstyle\sum_{i=1}^{n} \mathrm{Curv_i}) \cdot (r_i - r_{k-1}) \quad \text{for } (k = 2 \text{ to } n\text{-}1) \text{ and if } \left(r_{k-1} \leq r_i \leq r_{k-1} + 0.5(r_k - r_{k-1})\right)$$

$$S3_i = A \cdot (\mathrm{Curv_i}/\textstyle\sum_{i=1}^{n} \mathrm{Curv_i}) \cdot (r_i - r_k) \quad \text{for } (k = 2 \text{ to } n\text{-}1) \text{ and if } (r_{k-1} + 0.5(r_k - r_{k-1}) \leq r_i \leq r_k)$$

$$S_i = S1_i + S2_i + S3_i \tag{2.109}$$

Note that the source term S includes a periodic Sinus function ($S1_i$) with two additional linear functions ($S2_i$ and $S3_i$). Third, the 1D elliptic PDE equation with the RHS source term is solved for the new curvature based point distribution solution. Source terms are updated (because of embedded r_i) at each iteration of the 1D elliptic solver until a minimum residual is reached. The general discretized form of the proposed 1D elliptic PDE is:

$$r_{i-1} - 2r_i + r_{i+1} = S_i \tag{2.110}$$

The central difference discretized form of the 1D elliptic equation is solved using Point SOR or ADI schemes. Note that ADI scheme simplifies to the tridiagonal matrix algorithm (TDMA) or Thomas algorithm solver for 1D problem, simply by defining j=constant and only solving in i [85]. The transformation from solution r to physical space (x,y) is done using CAD surface. The CAD surface is preliminary generated for the geometry surface using spline curve (an out-of-CAD representation). The cases providing verification and parametric study of the proposed

methodology are presented in Chapter 5 with additional details in Appendices A and B. Note that extension of the curvature control algorithm to 3D grids would require projection on a surface (u,v). This would require advanced geometry representation and are outside the scope of this thesis.

2.2.1.2 Field grid generation algorithm

The application of ICEM-CFD commercial tool and the developed in-house grid generation tool NSGRID with the blended elliptic method and the solution schemes are described here.

ICEM CFD mesh generation tool (commercial)

Through a study performed at Polytechnique Montreal, a fully automated grid re-generation process using the ICEM-CFD commercial mesh generation package was developed within the multi-time steps framework of CANICE2D-NS [20, 42]. The grid generation procedure includes a preliminary algebraic grid generation and smoothing of the grid using an elliptic operator including different control functions to improve stretching, orthogonality and clustering of the grids. A standard elliptic smoothing technique using two background and foreground control functions is used to generate the multi-block structured grids around the ice shapes. The background control functions such as Middlecoff-Thomas influences the interior grid points and modifies the domain clustering [73]. The foreground control function such as Steger-Laplace or Steger-Sorenson influence the orthogonality and grid spacing control on the boundaries [74]. Blending of these methods within the grid smoothing process results in control of the quality of the grids over the entire domain. The number of foreground smoothing iterations is increased for the leading edge block containing the ice grids to conform to the ice shapes. This technique proves a necessary step to provide meshes with non-crossing cells (or positive cell areas) when highly complex ice shapes are presented (see Figure 2.19). Grid parameters such as wall distance and number of elements were selected based on analysis of several test cases considering trade-offs between computational cost, accuracy and robustness. Relevant additional details about the grids are given in Chapter 4.

Although this approach has been used for the framework of icing, it is not firmly robust. As seen in Figure 2.19, the smoothed multi-block grid visually contains poor grid qualities areas such as concave region clustering, block edge clustering extension, grid smoothness and stretching ratio. Also it is further observed that the orthogonality and wall spacing have not been retained well for complex ice grid problems. Automation of the grid meshing is also a time

consuming process. The blended elliptic smoother implemented in ICEM has not been shown to handle complex domain grids. Note that grid generation in 3D (considering only algebraic grid generation and quality improvements) is a very time consuming process using ICEM CFD.

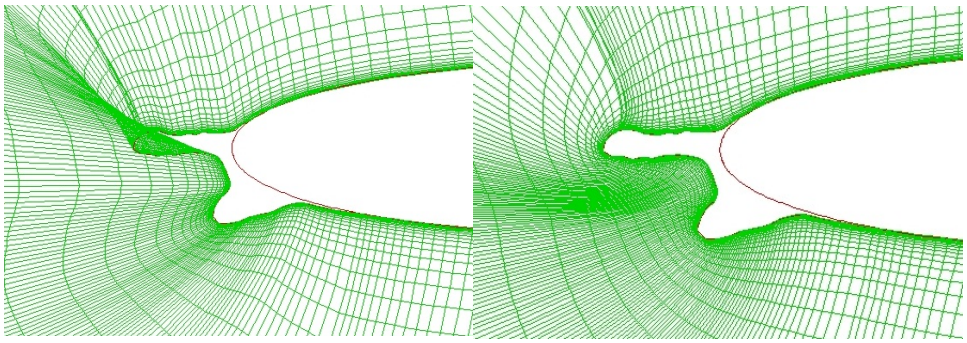

Figure 2.19: Mesh around an iced airfoil, algebraic (left) and elliptic smoothing (right).

NSGRID elliptic mesh generation tool (in-house)

PDE methods, such as elliptic Poisson equation, are more complex than algebraic methods but provide means to generate high quality grids by selection of control functions [21-23]. The algorithm performs an iterative procedure to generate the desired grid using algebraic grids as initial solution. Through the use of Laplacian operator, the grids metrics especially grid spacing and orthogonality close to wall are not retained as needed for RANS flow computations. Figure 2.20 shows a Laplacian operator solution around a cylinder, where one can see the non-uniform first later wall spacing at $\theta = 0^{\circ}$ and $\theta = 90^{\circ}$.

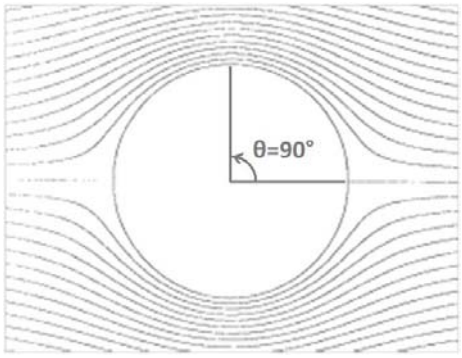

Figure 2.20: Grid spacing changes (curvature undesirable effects of Laplacian operators) [56].

To overcome this issue, a number of control functions and mapping transformation were examined [74, 84]. Considering Cartesian coordinates (x,y) for physical space and curvilinear coordinates (ξ,η) for computational space ($0 \leq \xi \leq \xi_{max}$ and $0 \leq \eta \leq \eta_{max}$), points one-to-one mapping can be done from physical domain to computational domain and reverse. Figure 2.21 shows the mapping for O-type and C-type mesh around an airfoil.

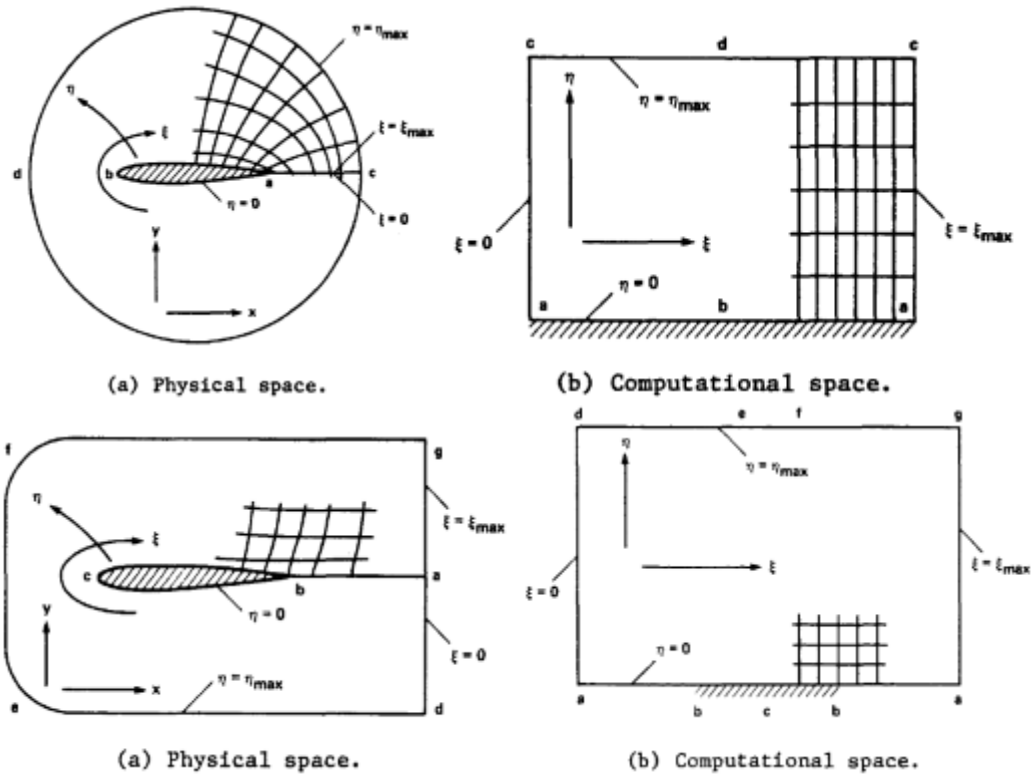

Figure 2.21: Mapping for O-type (up) and C-type (down) mesh around an airfoil [55].

Considering $\xi=\xi(x,y)$ and $\eta=\eta(x,y)$, one need to find the mapping equations to satisfy the Poisson equation [23]:

$$\xi_{xx} + \xi_{yy} = P \tag{2.111}$$

$$\eta_{xx} + \eta_{yy} = Q \tag{2.112}$$

where the terms P and Q are the forcing functions (control functions). Different choices for these values result in different grid characteristics. Using transformation equations, the Poisson equation can be written as [74]:

$$\xi_x = y_\eta/J, \quad \xi_y = x_\eta/J, \quad \eta_x = y_\xi/J, \quad \eta_y = x_\xi/J \tag{2.113}$$

$$J = x_\xi y_\eta - y_\xi x_\eta \tag{2.114}$$

$$(x_\eta^2 + y_\eta^2)x_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta)x_{\xi\eta} + (x_\xi^2 + y_\xi^2)x_{\eta\eta} = -(x_\xi y_\eta - y_\xi x_\eta)^2(Px_\xi + Qx_\eta) \tag{2.115}$$

$$(x_\eta^2 + y_\eta^2)y_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta)y_{\xi\eta} + (x_\xi^2 + y_\xi^2)y_{\eta\eta} = -(x_\xi y_\eta - y_\xi x_\eta)^2(Py_\xi + Qy_\eta) \tag{2.116}$$

These transformed equations are type of quasi-linear equations, which need to be linearized in order to be solved efficiently.

The choices for control functions here are based on Sorenson (RLS) (Section 2.1.3.7), Spekreijse (SPS) (Section 2.1.3.8), and the developed blended (RLS-SPS-Para) method. In the proposed blended method (RLS-SPS-Para) [41], the different approaches source terms are added with weight factors and decay functions (optional):

$$P(\xi,\eta) = f_{RLS}.P_{RLS} + f_{SPS}.P_{SPS} + f_{Para}.P_{Para} \quad (2.117)$$

$$Q(\xi,\eta) = f_{RLS}.Q_{RLS} + f_{SPS}.Q_{SPS} + f_{Para}.Q_{Para} \quad (2.118)$$

in which f_{RLS}, f_{SPS}, and f_{Para} are the weight factors for Sorenson, Spekreijse (with parametric space generated from algebraic grid normalized arclength), and Spekreijse (with parametric space generated from parabolic grid normalized arclength), respectively. The algebraic grid is generated using TFI method [24]. The paparabolic grids are generated using the approaches described in Sections 2.1.3.5. P_{RLS} and Q_{RLS} are the Sorenson approach source terms. P_{SPS} and Q_{SPS} are the Spekreijse approach source terms, where the parametric grid is defined by the generated algebraic grid. P_{Para} and Q_{Para} are the Spekreijse approach source terms, where the parametric grid is defined by the generated parabolic grid. Optional decay functions such as exponential, linear, or periodic functions, are also implemented to control the effects of parabolic grid source terms (P_{Para} and Q_{Para}) on the elliptic grid equations. Note that changes in the sources weight factors can result in changes in the stability of the elliptic solution algorithm.

The Spekreijse (SPS) source terms in the blended approach are either P_{SPS-SC} or $P_{SPS-SCO}$. The P_{SPS-SC} is the Spekreijse source term of spacing and curvature, that uses only the parametric space generated by algebraic grid. The $P_{SPS-SCO}$ is the Spekreijse source term of spacing and curvature with additional orthogonality. The orthogonality is achieved by elliptically smoothing the parametric space (with orthogonality boundary condition and algebraic parametric space as the initial solution). Adding Spekreijse orthogonality source term ($P_{SPS-SCO}$) results in a more relaxed problem and a better grid (for smoothed geometries), but a higher computation time because of additional elliptic solution on parametric space and also has difficulty at sharp convex corners. In the remainder of the paper, only the P_{SPS-SC} (named as P_{SPS}) is used for the proposed blended approach.

In the proposed blended approach, the methodology can be summarized as follows. First, generating the algebraic grid and its transformed parametric space. Second, generating the

parabolic grid (or the parabolic/algebraic grid) and its transformed parametric space. Third, computing the 2D elliptic transformed Poisson equation with the proposed blended source terms. The initial solution of the iterative solver can either be the algebraic grid or the parabolic grid. The RLS control functions are computed at each iteration. The SPS and Para control functions are computed only at the first iteration and held fixed throughout the entire grid computation. The decision of the weight factors are based on two main criteria: the elliptic PDE solution scheme stability and the grid quality criteria decided by the user. The value of each weight factors can be chosen between 0 to 1. The approaches validation and discussions are presented in Chapter 5 and additional details are in Appendices A and B.

Solution method

Linearization

Successive Substitution technique (or Simple Iteration technique) can be used to solve the nonlinear algebraic system [85]:

$$A(U)U = B(U) \tag{2.119}$$

Using this technique, the nonlinear system can be simplified to:

$$A(U^n)U^{n+1} = B(U^n) \tag{2.120}$$

where n=0,1,2,… is the iteration number.

Using the Successive Substitution technique, the transformed Poisson equation simplifies to:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2\left(Px_\xi + Qx_\eta\right) \tag{2.121}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2\left(Py_\xi + Qy_\eta\right) \tag{2.122}$$

$$\alpha = {x_\eta}^2 + {y_\eta}^2 \qquad \gamma = {x_\xi}^2 + {y_\xi}^2 \qquad \beta = x_\xi x_\eta + y_\xi y_\eta \tag{2.123}$$

where the coefficients α, β, γ, J are computed using the previous iteration solution, explicitly.

The Newton-Raphson technique has also been applied to linearize the problem of Poisson equation. Assuming the form of the equation is defined as follows [24]:

$$R = B(U) - A(U)U \tag{2.124}$$

$$\frac{\partial U}{\partial t} = R = 0 \tag{2.125}$$

By simplifying the left hand side, the implicit form of the equation can be written as:

$$\frac{U^{n+1} - U^n}{\Delta t} = R^{n+1} \tag{2.126}$$

Using the Taylor series expansion for the right hand side in n+1:

$$R^{n+1} = R^n + \frac{\partial R}{\partial U}\Delta U \tag{2.127}$$

The delta form of the equation simplifies to:

$$\left[\frac{I}{\Delta t} - \frac{\partial R}{\partial U}\right]\Delta U = R^n \tag{2.128}$$

where R^n is the computed residual at the time step n. To improve the stability of the iterative scheme, the Jacobian term $(\frac{\partial R}{\partial U})$ is active if $(\frac{\partial R}{\partial U} < 0)$, otherwise it is set to zero.

Solution of system

The linearized transformed Poisson equation by the Successive Substitution technique is solved using the approaches: Point-Jacobi (explicit), Point Gauss-Seidel (explicit), Point SOR (explicit), Line SOR (implicit) and ADI (implicit). The linearized transformed Poisson equation by the Newton-Raphson technique has been solved using the approaches Point-Jacobi (implicit) and ADI (implicit) [85, 41].

As anticipated, the form of linearization plays a noticeable role in the convergence of the implicit and explicit approaches. Numerical experimentation shows that ADI shows better stability and convergence using Newton-Raphson technique than Successive Substitution technique for linearization of Poisson equation.

Full Multi-Grid approach

A Full Multi-Grid (FMG) approach has been applied to the grid generation solver to reduce the computation time, especially for ice grid generation extension in 3D [86, 87]. Assuming the form of the equations is defined as below:

$$LU = F \tag{2.129}$$

where L is the operator, U is the solution, and F is the right hand side.

The recursive nonlinear FMG approach is defined by the following steps:

1. Solve the equation on the fine grid (k) with defined sweeping times (S_1) and initial solution (U_{old}^k):

$$U^k = \text{Solve}^{S_1}(U_{old}^k, L^k, F^k) \tag{2.130}$$

2. Compute the residual on the fine grid (k):

$$R^k = F^k - L^k U^k \tag{2.131}$$

3. Restrict the residual and solution from the fine grid (k) to the coarse grid (k+1):

$$R^{k+1} = I_{k+1}^{k} R^{k} \tag{2.132}$$

$$\bar{U}^{k+1} = I_{k+1}^{k} U^{k} \tag{2.133}$$

4. Solve the error (e) on the coarse grid (k+1) with defined sweeping times (S_2), initial solution ($\bar{U}^{k+1}$) and right hand side (R^{k+1}):

$$e^{k+1} = \text{Solve}^{S_2}(\bar{U}^{k+1}, L^{k+1}, R^{k+1}) \tag{2.134}$$

5. Prolong the error (e) from the coarse grid (k+1) to the fine grid (k):

$$e^{k} = I_{k}^{k+1} e^{k+1} \tag{2.135}$$

6. Correct the solution on the fine grid (k) using the prolonged error (e^k):

$$U_{new}^{k} = U^{k} + e^{k} \tag{2.136}$$

7. Solve the equation on the fine grid with defined sweeping times (S_3) and initial solution (U_{new}^{k}):

$$U^{k} = \text{Solve}^{S_3}(U_{new}^{k}, L^{k}, F^{k}) \tag{2.137}$$

where R is the residual, I_{k+1}^{k} and I_{k}^{k+1} are the restriction and prolongation operators, respectively. This recursive procedure is repeated until a residual criterion is reached. An extensive description of the method and the results are discussed in Chapter 5 and Appendices A and B.

2.2.2 RANS flow simulation

Flow field parameter solution for the icing simulation is obtained by Navier-Stokes based solvers [8, 14], replacing the traditional potential inviscid flow integrated with viscous boundary layer calculation. The effects of viscous flow on the icing simulation augment as the complexity of ice shape increases especially for glaze ice cases. Therefore, research is moving towards using RANS modeling. The main difficulty of RANS based methods comes from the point of ice grid generation complexity and high computation times. Here we briefly discuss the RANS solvers and roughness models used in the studies for the coupling to the framework of aero-icing simulation, CANICE-NS code.

2.2.2.1 RANS flow solvers

There are number of RANS based solvers that have been linked to the CANICE-NS code: P/SMB [43], NSMB [44], and NSCODE [45]. The first RANS based flow solver coupled to CANICE-NS is P/SMB, which is a parallel or serial Multi-Block three dimensional Euler/Navier-Stokes flow solver developed by Badcock et al. [88, 89]. P/SMB3D is a cell-

centered finite-volume code and can be applied for compressible steady and unsteady flow simulations. A wide range of turbulence models such as k-ω, k-ε, k-ω-SST, Spalart-Allmaras, hybrid (DES based on Spalart-Allmaras) are available and validated. Parallel processing version of the code is configured to decompose the structured multi-block grids solution on different processors using MPI and PVM. P/SMB3D has been used for a wide variety of aerospace applications [90]. The rough-wall extension treatment, important for icing simulation, is not implemented in P/SMB. The code is integration to CANICE2D-NS for multi-time step icing simulation and effects analysis [39].

The second code linked to CANICE-NS is NSMB, a three dimensional Euler/Navier-Stokes finite-volume flow solver developed by J. B. Vos et al. [26, 44, 91]. The code uses multi-block structured grids and can be applied for parallel or serial processing. A data base library "Memcom" is available to increase the speed of the data access. The turbulence models implemented in NSMB are: algebraic Baldwin Lomax and Granville, Spalart-Allmaras, k-ε, k-ω-SST, and others. Also, a number of rough wall treatment extensions such as ONERA extension of Spalart-Allmaras, and Knopp and Wilcox extension of k-ω are available to simulate the effect of surface roughness in the flow computation. The code includes pre-conditioning option for low Mach number flows and has a Full Multi-Grid method to reduce the computation time. NSMB can be applied to steady and unsteady flow simulations [92, 93]. NSMB is integrated to CANICE2D-NS for RANS based single and multi-time step icing simulation with the analysis of roughness effects. Results, validation and comparison are published and discussed in Chapter 4 [20, 31, 63].

The third flow solver integrated with CANICE-NS is NSCODE, a finite volume two dimensional multi-block Euler/Navier-Stokes based solver with Full Approximate Storage Multi-Grid developed by the laboratory of prof. Laurendeau [31, 94]. Available turbulence models include the Spalart-Allmaras, k-ω-SST and γ-Rθ equations. The code incorporates the wall treatment roughness model of Boeing which is implemented and validated within the Spalart-Allmaras turbulence model. NSCODE2D is capable of steady and unsteady flow analysis and incorporates a Chimera method. The code is verified and validated on a variety of steady and unsteady case studies [45, 63]. Results of CANICE-NS couple with NSCODE2D are discussed in Chapter 6 [13, 31, 46, 63].

2.2.2.2 Roughness modeling capabilities

Surface roughness is one of the main parameters influencing the surface skin friction and heat transfer, and has a major influence in the ice accretion process. Roughness values are usually assumed to be constant on the surface, but in reality, ice shape roughness varies in space and time [20, 29, 37].

The rough wall treatment implementations (roughness modeling in RANS solvers) in Spalart-Allmaras and k-ω models are examined. There are two extensions for the Spalart-Allmaras roughness implementation: ONERA and Boeing [27]. In the two equations k-ω turbulence model, two rough wall models are addressed: Wilcox and Knopp methods [26, 28]. The Boeing extension rough wall treatment of Spalart-Allmaras is implemented in NSCODE. ONERA extension rough wall treatment of Spalart-Allmaras and k-ω model both Wilcox and Knopp rough wall models are implemented in NSMB code. Table 2.1 shows the rough wall treatment models available in each RANS solver. The roughness models formulations are cited in Appendix A.

Table 2.1: RANS solvers roughness models info.

	S-A		k-ω	
Code	**Boeing**	**ONERA**	**Wilcox**	**Knopp**
P/SMB	-	-	-	-
NSMB	-	yes	yes	yes
NSCODE	yes	-	-	-

2.2.3 RANS solution integration to CANICE2D-NS framework

Following the convention used within NASA's LEWICE code [12], the CANICE2D-NS framework couples the Navier-Stokes solver in two modes. Mode 1 makes use of the flow field solution obtained from RANS solver (density, velocities, pressure), instead of the panel method solution to determine the particle trajectories, impingement efficiency, and boundary layer parameters. Mode 2, in addition to mode 1, uses the Navier-Stokes skin friction coefficients in the thermodynamic module. Note that all the developments results presented here are based on Mode 2 integration.

The RANS flow solution integration is done using a volume/area weighted interpolation as follows (Figure 2.22):

$$P = (P_1A_1 + P_2A_2 + P_3A_3)/A \tag{2.138}$$

where A is the area of triangle 1-2-3 and P is the flow parameters at point 4.

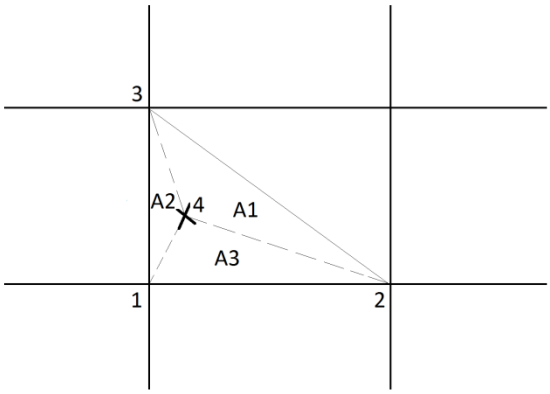

Figure 2.22: Area weighted interpolation for 3 points.

The position search also is calculated using these areas as follows:

$$|(A_1 + A_2 + A_3) - A| \leq e \tag{2.139}$$

where e is the threshold defined by user.

The surface parameters interpolation is done using a linear interpolation between 2 points. Since all RANS solutions are cell centered, a simple algebraic average is used to transfer cell-centered values to cell-vertex one. These functions were available within each RANS library.

The single stagnation point is defined based on the maximum pressure coefficient solution computed by the RANS solver around the leading edge. The stagnation point is the reference point for water runback direction assumption and ice thermodynamics computations.

The heat transfer coefficients are calculated within the thermodynamic module through empirical methods relating to the skin friction and roughness values [95]. The heat transfer coefficient for the turbulent region is defined by (eq. 2.21).The Stanton (St) and roughness Stanton number (St_k) are defined by the equations (2.19, 2.20). The roughness value is defined by the user as an uniform or non-uniform value, although only constant roughness is used in the thesis. The boundary-layer edge velocity from the Navier-Stokes solution is determined from the isentropic relation for the wall pressure.

CHAPTER 3

CONSISTENCY OF THE ARTICLES

3.1 Context

One of the major concerns in flight safety for flying in any weather conditions is ice accretion effects problems. Ice can be accreted on aircraft body surfaces such as wings, engines intakes, control surfaces, and instruments. One destructive effect is related to the complex ice formation on the aircraft lifting wings. As complexity of ice increases, shape or thickness, the flow disturbances on the wing surface result in higher reduction of aerodynamic performances, potentially leading to flight accidents. Understanding the physic of the complex ice accretion process and development of an efficient ice accretion simulation tool is one of the main objectives toward the analysis, control and reduction of flight accident due to icing effects. The main objective of the project is the development of the Navier-Stokes based icing simulation tools to perform efficient, automated, quasi-steady (multi-time steps) icing simulation for prediction of complex 2D glaze ice shapes, with algorithms compatible with 3D extension.

To achieve this objective, first, the RANS code PMB3D was validated and coupled with the icing code CANICE2D. The panel method flow module in CANICE was deactivated and the flow solution (velocities and surface parameters) interpolated directly from the RANS solver solution. The traditional Messinger model and water runback simulation used in CANICE, are based on the assumption of single stagnation point which defines the water runback direction on the surface. The RANS flow, starting from field velocities were interpolated to the Lagrangian droplet trajectory module. The RANS surface skin friction solution is interpolated to the heat transfer computation module which is base on semi-empirical turbulence flow heat equation. It was discovered that mesh generation needs careful attention, since the panel method does not require a field mesh.

Using the ICEM CFD commercial tool, a multi-block structured template was prepared. To perform remeshing process, the replay control developed within ICEM-CFD was used, with additional grid smoothing on the airfoil surface blocks in cases of ice growth. Finally for the multi-time steps icing simulation, a batch process was developed to repeatedly run the grid

generation, RANS solver and icing simulation CANICE2D-NS for defined number of time steps until it reaches the total ice accretion time and finally to post process the results.

The results of these developments in addition to iced airfoil aerodynamic performances analysis were presented at the CFDSC conference in May 2012 [39]. The paper presents results of iced airfoil performance in comparison to experimental data using the newly developed two-dimensional ice accretion simulation code, CANICE2D-NS. The framework is capable of performing multi-layer simulation and the flow analysis and performance prediction of iced airfoils. The results show good agreement with experimental data both in terms of predicted ice shapes, as well as aerodynamic analysis of predicted and experimental ice shapes including surface pressure distribution, lift and drag comparisons.

Since PMB did not handle surface roughness, it was decided to couple the NSMB 3D RANS solver. Not only it contain surface roughness turbulent model, it moreover had a chimera capability, useful in the treatment of complex geometries. The NSMB3D solver was coupled to the framework of CANICE2D-NS with a further refined ICEM-CFD remeshing replay control to perform multi-time steps icing simulation. Here focus was the effects of surface roughness on ice accretion simulation. The roughness value is also incorporated in the semi-empirical heat transfer equation in CANICE thermodynamic module. A study was performed to achieve time independent icing solution by increasing the number of time steps in multi-time steps icing simulation up to 160. Results show the necessity of icing simulation with large number of time steps, of order of 20-40. This is significant, since most icing frameworks used 1-5 steps. This was possible because of our mesh regeneration procedure. These results were published in AIAA Journal of Aircraft (June 2013), attached in Chapter 4 [20]. The results demonstrates the benefits of the new framework in predicting ice shapes and aerodynamic performance parameters as well as iced airfoil surface pressure coefficients. Finally, the convergence of the quasi-steady algorithm is verified and identifies the need for an order of magnitude increase in the number of multi-time steps in icing simulations.

The newly developed RANS solver NSCODE2D, developed within the research laboratory of prof. Laurendeau at Polytechnique Montreal, was also coupled with the framework of CANICE2D-NS. Here the focus is to improve the RANS icing computation times through the use of NSCODE solver capable of Full Multi-Grid (FMG) computation speed up. For the purpose of RANS multi-time steps simulation with high number of time steps (such as 200), the

flow computation times for 2D problem was improved by a factor of 10 in number of non-linear iterations compared to the use of NSMB. NSCODE also is capable of roughness computation through the use of wall treatment roughness model Boeing implemented in Spalart-Allmaras turbulence model. It also includes variety of turbulence models such as k-ω and γ_{r_θ}, which have been incorporated in the ice accretion simulation and analysis. NSCODE can perform calculations on chimera meshes. The efficiency of the J-multigrid approach to solve the flow equations on complex iced geometries is demonstrated. Finally, results on up to 160 quasi time-steps calculations are presented and analyzed. These developments and analysis were published in CASI (May 2013) and AIAA Scitech conference (June 2013) [63, 31].

Up to this stage of the project, ICEM-CFD was used for the multi-time steps icing framework of CANICE2D-NS. The multi-block structured grids generated by ICEM CFD are mainly algebraic grids. The elliptic smoothing approaches implemented in ICEM do not have sufficient properties to overcome the problem of very complex ice shapes grids. It was observed that the grid quality properties such as spacing, orhtogonality, skewness are not retained when using the various available smoothers (such as elliptic blended methods: Steger-Sorenson or Thomas-Middlecoff) on complex ice forms. Also ICEM needs a vast amount of time and user input to configure a suitable grid for a specific case study. This process is highly time consuming as the user need to configure the software input to address a number of issues such as the geometry and blocking definition, block edges modifications, edges point distribution, the smoother application on each block and its proper iterations, to obtain a user defined proper grid for an experimental or numerical icing airfoil surface case study. The automation is another issue using ICEM-CFD meshing tool which is based on the replay control batch file that records all the process of grid generation by the user for a specific case study. For the 3 dimensional icing problems, ICEM-CFD grid generation issues would increase dramatically.

These difficulties let to develop a meshing tool specific for ice shape grid generation, capable of generating high quality grids for the problem of complex glaze ice. A new PDE curvature based surface mesh generation algorithm was developed handling surface meshing sharp corners, concave and convex regions. A new elliptic blended method was developed to properly generate high quality field grid for very complex ice shapes. The meshing tool provides a multi-block capability. Variety of grid generation methods such as algebraic, parabolic, elliptic and blended parabolic/elliptic were implemented. Different choices of elliptic smoothing methods such as

Sorenson, Spekreije, boundary orthogonal method and blended methods were validated and compared. The algorithm has a Full Multi-Grid capability, and particular attention was paid to the choice of smoother. It also includes Point Jacobi, Point Gauss-Seidel, Point and Line SOR, ADI, including decoupled i or j or i and j. The code is automated and coupled within the framework of multi-time steps icing codes such as CANICE2D-NS and NSCODE-ICE [34]. The code has been validated through generation of high quality grid for very complex experimental ice cases and the aerodynamic performance analysis. These grid generation code developments and results were presented at the AIAA conference January 2015 and in AIAA Journal (Accepted on September 2015), attached in Chapter 5 [96, 41].

The CANICE2D-NS framework, coupled with NSCODE2D and NSGRID2D, were validated through a vast number of icing case studies at Bombardier advanced aerodynamics department including benchmark NATO cases. The aim is study the robustness of the framework based on the number of successful computation compared to the total number of test runs. The results demonstrate the benefits of the new framework in predicting ice shapes and aerodynamic performance parameters. These developments, results and discussions are presented in Chapter 6. A Technical report was delivered to Bombardier Aerospace (October-November 2014) [13]. These results led to a joint Polytechnique Montreal/Bombardier publication at CASI conference (May 2015) [46].

The 2D framework also includes an Eulerian droplet solver (developed at Polytechnique Montreal). Furthermore, the developed grid tool NSGRID2D was extended to 3D and coupled with the 3D icing simulation code NSMB3D-ICE (developed by Universite de Strasbourg and Polytechnique Montreal) to develop the framework of 3D multi-time steps icing simulation code NSMB3D-ICE/NSGRID3D. The resulted developments were published in SAE conference (June 2015) [47]. These developments are excluded from the thesis objectives, but nevertheless show the feasibility of the proposed methodologies for 3D problems.

CHAPTER 4 ARTICLE 1:

QUASI-STEADY CONVERGENCE OF MULTI-STEP NAVIER-STOKES ICING SIMULATIONS

Article published in AIAA Journal of Aircraft (July-August 2013), written by:

Kazem Hasanzadeh Lashkajani, Eric Laurendeau, Ion Paraschivoiu

École Polytechnique de Montréal

Abstract

A newly developed two-dimensional ice accretion and anti-icing simulation code, CANICE2D-NS is presented. The method is used to predict iced airfoil shapes and performance degradation with a multi-step approach. A Multi-Block Navier-Stokes code, NSMB, has been coupled with the CANICE2D icing framework, supplementing the existing panel method based flow solver. Attention is paid to the roughness implementation within the turbulence model, and to the convergence of the steady and quasi-steady iterative procedure. The new coupling allows fully automated multi-layer icing simulation while also permitting flow analysis and performance prediction of iced airfoils. Effects of uniform surface roughness in quasi-steady ice accretion simulation are analyzed through different validation test cases. The results demonstrates the benefits and robustness of the new framework in predicting ice shapes and aerodynamic performance parameters as well as iced airfoil surface pressure coefficients. Finally, the convergence of the quasi-steady algorithm is verified and identifies the need for an order of magnitude increase in the number of multi-time steps in icing simulations.

4.1 Introduction

An iced airfoil, whether on an aircraft or wind-turbine, can show significant reduced lift and increased drag properties which stalls at lower angle of attack compared to its clean shape. The flow physics over the iced airfoil, especially around glaze ice horns, is quite complex and depends on the flow regime, as described in [4, 6]. For instance, at low angle of attacks, the flow can display separated flow regions behind the horns reattaching on the upper surface. At higher angle of attacks, the flow might separate entirely aft of the horns. In either case, the effects on

the airfoil characteristics can be significant. Today's aircraft industry relies on three methods to determine the icing effects on the aerodynamic performances of aircraft: flight testing, wind tunnel testing and computational fluid dynamics (CFD) simulations. The methods differ in costs, simulation constraints for realistic conditions and accuracy. The paper concentrates on the CFD approach, in particular the multi-step procedure for prediction of complex ice shapes stemming from long flight times and their influence on the aerodynamic characteristics of aircraft.

In the field of numerical ice accretion simulations, the traditional method [8-10] contains an inviscid flow solver (panel method based) to obtain the flowfield. The potential solution is then used to determine the water droplet trajectory and droplet impingement distribution via the Lagrangian approach. A boundary layer formulation, integral or differential, is then used to determine the skin friction, the local heat transfer coefficient and the near-body flow characteristics. Finally, a Messinger model [11] is used for ice accretion thermodynamic analysis. These algorithms can become inaccurate or may even fail to provide a solution when the complexity of the ice formed on the surface increases, especially on glaze ice with sharp horns. As viscous effects become more important, modeling of the flow via the panel method proves inadequate [52]. Additional difficulties arise from handling of multiple stagnation point within the boundary-layer parabolic marching scheme as well as increased discretization errors with paneling deformations. Improvements in the field of aircraft icing simulation has come from integration of Navier-Stokes based solver to model the flow field via the Reynolds Averaged Navier-Stokes equations (RANS) [12, 14]. The main difficulty of RANS based icing simulation comes from grid generation, especially for complex ice shapes. Computation time and memory usage are also significantly increased. Research is thus needed to address the various difficulties associated to the use of RANS based methodologies.

In this paper, the introduction of a RANS solver within an icing framework is examined. Focus is on the development of a fully automated multi-step coupling procedure, capable of analyzing long ice accretion accumulation times (45 minutes) in a quasi-steady formulation. The complexity of grid generation within the multi-step procedure to account for large deformations with concave/convex topologies is addressed. The choice for automation of the meshing process via mesh regeneration is justified. Effects of icing roughness modeling, important in rough rime ice accretion conditions are examined. Finally, convergence studies are performed to validate the approach. The paper concludes that quasi-time step convergence must be achieved when

performing long ice accretion simulations times. Only then can modeling errors be properly assessed.

4.2 Methodology

4.2.1 Framework

The icing/anti-icing simulation code CANICE2D has been developed at École Polytechnique de Montréal as part of a collaborative R&D activities funded by Bombardier Aerospace and the Natural Sciences and Engineering Research Council of Canada (NSERC) over fifteen years [5, 15-18, 67, 71]. The version of CANICE presented here is the research version which differs from the version used at Bombardier Aerospace named CANICE-BA in the literature [19], but contains the same four basic modules: external flow simulation, droplet trajectory and local catch efficiently calculation, surface ice/water interface thermodynamic balance and ice accretion, and finally hot air anti-icing simulation (see Figure 4.1).

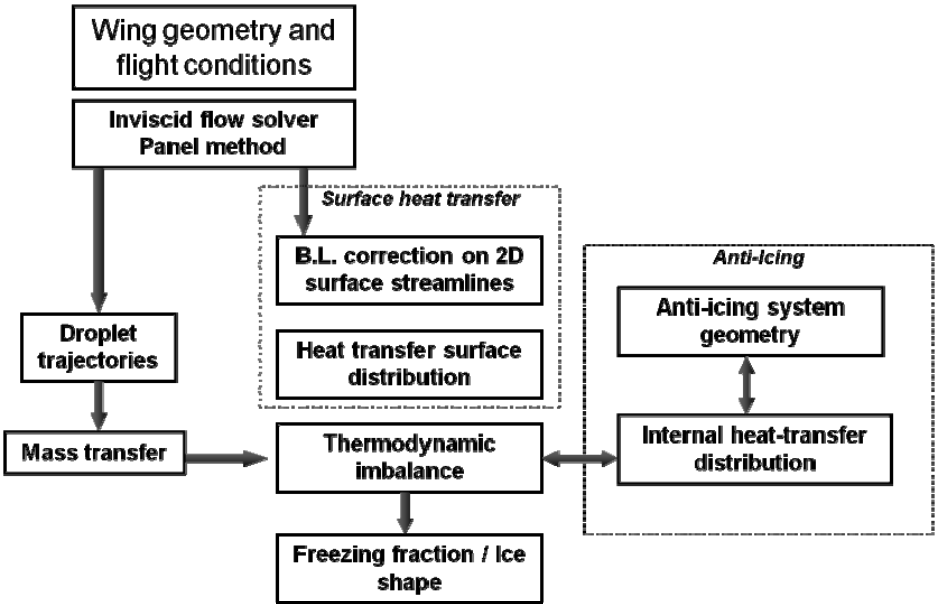

Figure 4.1: CANICE code structure.

The code can be applied for single and multi-element simulation, as discussed in [18]. The flow solution is obtained by the Hess and Smith panel method approach, used for droplet trajectory and boundary layer calculations. An integral boundary layer formulation is used for simulation of viscous effects. The first version of CANICE2D used the integral laminar flow method of Thwaites, the laminar to turbulent transition model of Michel and turbulent boundary

calculation using the lag-entrainment method [71]. The effect of roughness on the boundary layer model, the transition criteria and skin fiction/heat transfer calculations for laminar and turbulent regions are implemented using roughness based empirical relations. Conceptually, roughness is based on the equivalent sand-grain roughness height, which is determined by empirical methods [5].

Droplet trajectories are simulated using a Lagrangian approach, which solves the equation of motion of the water droplets for the defined time intervals using a fourth-order Runge-Kutta scheme. It interpolates the solution of the panel method potential flow solver to estimate the drag force on the spherical water droplets along with gravity and buoyancy forces to finally compute the water droplet trajectories and impingement limits on the airfoil surface [2]. The droplet impingement distribution is used to calculate the water droplet local catch efficiency on the surface which is then used to estimate the droplet mass flow rate impinging on the surface panels.

Using the calculated local convective heat transfer and water droplet impingement flow rates on the surface panels along with runback water mass rate from neighboring panels, the mass and energy balance is applied to the control volumes to solve for the amount of super-cooled impinged water droplet converted to ice mass. The Messinger model is used to define the type of ice surface (wet, dry rime or wet glaze), freezing fraction (fraction of ice mass to the entering total mass flux of the control volume), and surface temperatures [11].

To take into account the hot air anti-icing heat flux, the anti-icing module solves the internal heat transfer coefficient using an empirical correlation related to impinging jet on flat plate. The correlation is function of an average Nusselt number based on the jet parameters such as jet Reynolds number, nozzle to surface distance, and nozzle width [16]. Using the calculated internal hot-air local heat transfer coefficients and conduction through the thin leading edge skin (neglecting temperature variations across the plate thickness), the wall heat transfer rate is calculated and integrated in the thermodynamic balance via an iterative process. The final ice geometry is updated by calculating the ice height growth at the center of each panel, interpolating and smoothing the panel connectivity's on the edges using panel size and angle criterion [5].

A number of modifications have been added to CANICE2D, such as compressibility effects to the potential inviscid flow solution, multi stagnation point smoothing, roughness prediction,

shear-driven water runback model, ice shape updates and smoothing, water droplet splash and shading effects on the droplet local catch efficiency and water loss, etc. These were added to increase simulation accuracy as well as robustness. The CANICE2D icing framework has been validated through NATO/RTO exercises (see Figure 4.2) [9], which shows the need to increase the fidelity of the simulations especially in glaze ice horns conditions.

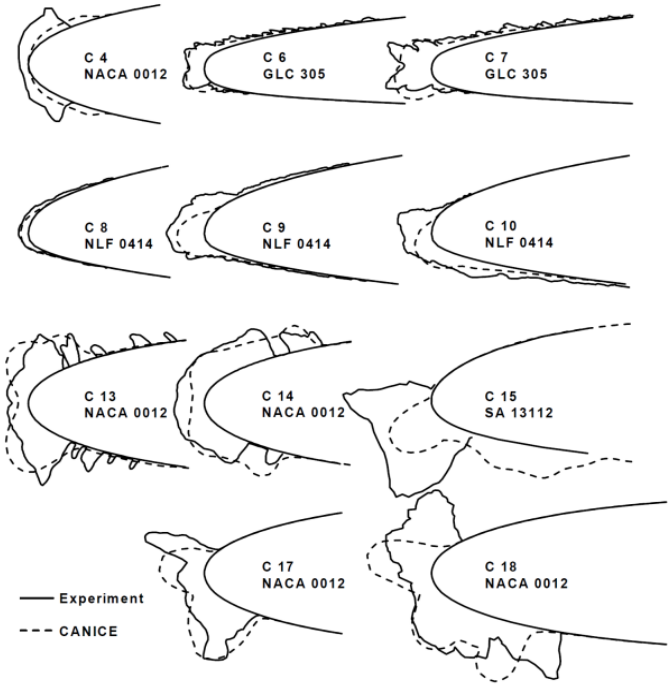

Figure 4.2: CANICE2D results and comparison for NATO/RTO exercise case studies [9].

4.2.2 Navier-Stokes solver

The Navier-Stokes flow solver NSMB3D, a finite volume three dimensional multi-block Euler/Navier-Stokes flow solver developed by Vos et al. [91, 97, 98] was incorporated within the CANICE icing framework. The code is being developed by a group of researchers across Europe and now in Canada, and benefits from state-of-the-art advances in computational techniques: low Mach number preconditioning, Full Approximate Storage Multigrid, dual time stepping marching schemes for steady and unsteady flows simulations, Arbitrary Eulerian-Lagrangian formulation suitable for aero-elastic calculations, and parallel implementation. Available turbulence models include the widely used Spalart-Allmaras and k-ω-SST models. The chimera approach is also implemented [92, 93]. Validations have been presented at the AIAA

validation workshop series, namely drag predictions [99], high-lift [100] and aero-elastic [101] and in a number of European framework program reports [102].

4.2.3 Roughness modeling

Surface roughness plays a role in the determination of the surface skin friction and heat transfer coefficients. These can significantly alter the ice accretion process. One assumption is that the roughness size in any direction is smaller than the boundary layer thickness. Surface roughness height is generally computed using empirical models such as the sand-grain model, which takes into account height and density distribution. However, in Navier-Stokes mode, there exist many models that have addressed surface roughness. The model of ONERA [27] has been considered in this study, although reference [37] uses Boeing extension.

The equivalent sand grain roughness height model is used to impose the wall roughness value. The effect of roughness is implemented within the S-A model by modifying the turbulent eddy viscosity in the near-wall region. One can relate the roughness height to changes in the velocity profile which then changes the wall skin friction. This implementation needs calibration by defining values of skin friction changes to the roughness value. For completeness, the Boeing and ONERA models are summarized.

Boeing's extension replaces the Dirichlet wall condition by a Neumann condition:

$$\frac{\partial \tilde{\nu}}{\partial n} = \frac{\tilde{\nu}}{d} \tag{4.1}$$

The wall distance is computed by:

$$d = d_{min} + d_0 \tag{4.2}$$

Where d_{min} is the grid distance to the nearest wall and d_0 is a shift which depends on the roughness value (k_s) by:

$$d_0 = exp\,(-8.5k)k_s \approx 0.03k_s \tag{4.3}$$

Changes in the wall distance cascades into the transport equation. This includes modifying the f_{v1} function in the Spalart-Allmaras model, for better prediction of smaller roughness, through a new definition of the χ term in S-A model [27]:

$$\chi = \frac{\tilde{\nu}}{\nu} + 0.5\,\frac{k_s}{d} \tag{4.4}$$

The Boeing model extension for rough wall has been implemented in the icing code FENSAP-ICE within the Spalart-Allmaras turbulence module and has been used for roughness

effects analysis. The effect of roughness is shown by the rime and glaze ice case studies simulated by FENSAP-ICE [37].

ONERA extension's (hereafter referred to as SA-ONERA) centers on defining a non-zero value for turbulent viscosity at the wall. To determine the transport quantity at the wall, the dimensionless form of the Spalart-Allmaras transport and momentum equation with viscosity ν and friction velocity u_τ becomes:

$$c_{b1}\tilde{S}^{+}\tilde{\nu}^{+} - c_{w1}f_w(\frac{\tilde{\nu}^{+}}{d^{+}})^2 + \frac{1}{\sigma}[\frac{\partial}{\partial y^{+}}\left(\tilde{\nu}^{+}\frac{\partial\tilde{\nu}^{+}}{\partial y^{+}}\right) + c_{b2}(\frac{\partial\tilde{\nu}^{+}}{\partial y^{+}})^2] = 0 \tag{4.5}$$

$$\frac{\partial u^{+}}{\partial y^{+}} - (u'v') = (1+\nu_t^{+})\frac{\partial u^{+}}{\partial y^{+}} = 1 \tag{4.6}$$

By imposing the boundary conditions $\tilde{\nu}^{+}$at the wall and far within the logarithmic region and solving the resulting non-dimensional transport equation, the solutions are obtained. The velocity profile included in $\tilde{S}$ is used to determine the velocity shift Δu^{+}. Changes in velocity profile can be related directly to the skin friction value on the wall. Similar to Boeing's extension, the wall distance also has been shifted to control the wall boundary value through a relation that combines the smooth wall distance d^{+}_{min} and imposed transport quantity value $\tilde{\nu}^{+}_{w}$ at the wall:

$$d^{+} = d^{+}_{min} + \frac{\tilde{\nu}^{+}_{w}}{k} \tag{4.7}$$

where k is Von Karman constant. Relating the normalized sand-grain roughness ($k_s^{+} = \frac{k_s u_\tau}{\nu}$) to the imposed wall value $\tilde{\nu}^{+}_{w}$ is done through the following correlations to close the problem:

$$\tilde{\nu}_w^{+} = (0.377ln(k_s^{+}) - 0.447)exp\left(-\frac{k_s^{+}}{70}\right) + 1.25710^{-2}k_s^{+}\left[1 - exp\left(-\frac{k_s^{+}}{70}\right)\right]$$
$$+max\left(0; log\left(\frac{k_s^{+}}{10}\right)\right)min\left(1; 1.36exp\left(-\frac{k_s^{+}}{250}\right); 25exp\left(-\frac{k_s^{+}}{100}\right)\right) \tag{4.8}$$

Increased roughness plays an important role in icing calculations as it augments the surface skin friction and heat transfer which results in changes in ice shapes growth. Note that only a constant roughness height specification is available in the current implementation for a given calculation step.

4.2.4 Coupling mode

Following the convention used within NASA's LEWICE code [12], the CANICE2D-NS framework couples the Navier-Stokes solver in two modes. Mode 1 makes use of the flow field solution obtained from NSMB (density, velocities, pressure), instead of the panel method solution to determine the particle trajectories, impingement efficiency, and boundary layer parameters. Mode 2, in addition to mode 1, uses the Navier-Stokes skin friction coefficients in the thermodynamic module. The heat transfer coefficients are calculated within the thermodynamic module through empirical methods relating to the skin friction and roughness values [95]. The heat transfer coefficient for the turbulent region is defined by:

$$h_{c,t} = St.\rho.u_e.c_p \tag{4.9}$$

The Stanton (St) and roughness Stanton number ($\mathrm{St_k}$) are defined by:

$$St = \frac{C_f/2}{Pr_t + \frac{\sqrt{C_{f/2}}}{St_k}} \tag{4.10}$$

$$St_k = 1.156[\frac{u_\tau k_s}{\nu}]^{-0.2} \tag{4.11}$$

Where $\mathrm{Pr_t} = 0.9$ is the turbulent Prandtl number for air. The roughness height $\mathrm{k_s}$ is defined based on the equivalent sand-grain roughness model, and the skin friction is related to the shear velocity $\mathrm{u_\tau}$ and the boundary-layer edge velocity $\mathrm{u_e}$:

$$u_\tau = u_e\sqrt{\frac{C_f}{2}} \tag{4.12}$$

The boundary-layer edge velocity from the Navier-Stokes solution is determined from the isentropic relation for the wall pressure.

4.2.5 Mesh generation

CFD modeling for icing simulations is impeded by the absence of a robust grid generation process which creates high quality Navier-Stokes mesh for severe mixed concave/convex glaze ice shapes. Automation of the grid generation process is another important aspect for performing multi-time steps during icing simulations. The process can iterate between the icing sequences via mesh regeneration or mesh movement (spring analogy [60], elastic [61], adjoin-based [30]). Although mesh movement is used by some researchers for icing simulations [12, 14], it has been found that, at least in 2D, the alternate approach of mesh regeneration provides increased

robustness. In the present study, a fully automated grid re-generation process using the ICEM-CFD commercial mesh generation package was developed within CANICE2D-NS. The grid generation procedure includes a preliminary algebraic grid generation and smoothing of the grid using an elliptic operator including different control functions to improve the stretching, orthogonality and clustering of the grids. A standard elliptic smoothing technique using two background and foreground control functions is used to generate the multi-block structured grids around the ice shapes. The background control functions such as Middlecoff-Thomas influences the interior grid points and modify the domain clustering [73]. The foreground control function such as Steger-Laplace or Steger-Sorenson influence the orthogonality and grid spacing control on the boundaries [74]. A blending of these methods within the grid smoothing process results in control of the quality of the grids over the entire domain. The number of foreground smoothing iterations has been increased for the leading edge block containing the ice grids to conform to the ice shapes. This technique proves a necessary step to provide meshes with non-crossing cells (or positive cell areas) when highly complex ice shapes are presented (see Figure 4.3). Grid parameters such as wall distance, number of elements, etc. were selected based on analysis of several test cases considering trade-offs between computational cost, accuracy and robustness. Relevant additional details about the grids will be given within each validation sections.

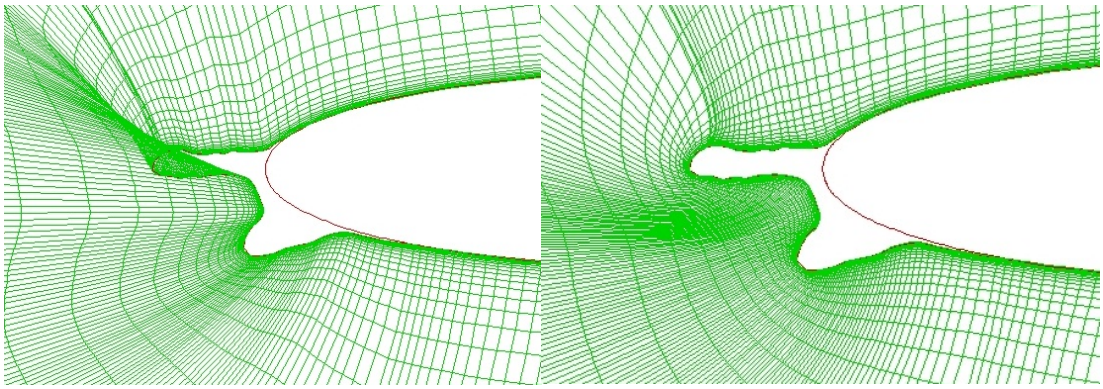

Figure 4.3: Mesh with algebraic (left) and elliptic smoothing (right) around ice shape.

4.3 Results and discussion

In this section, different simulation cases are presented with the objective to validate first the NSMB code then the CANICE2D-NS icing code by comparing results with known experimental data. The selected test cases are the smooth and rough flat plate, the NACA0012 and RAE2822 clean airfoil and several rime and glaze ice cases that are considered standard within the icing

community. All results in this paper using the NSMB solver have been run in mode 2 with the SA-ONERA rough wall treatment on grids generated with the elliptic grid generation package of ICEM CFD.

4.3.1 Validation of NSMB solver

4.3.1.1 Rough wall turbulent skin-friction over flat plate

The selected case for the ONERA rough wall model validation within the flow solver is the case of Hellsten and Laine [28] with the following specifications: chord of 1 m, Reynolds 5 Million, Mach 0.2, normalized roughness height K_s of 0.00005 (smooth surface equivalent), 0.0001, 0.00025, 0.0005, 0.001, 0.0015. The Cartesian grids have 42,000 nodes with a 1st layer wall spacing of 10^{-6} chord ($y^+<1$) and an expansion ratio of 1.5 in the normal wall direction (see Figure 4.4). Convergence levels reached 10^{-5} on the L2-norm of the density residual, as shown in Figure 4.5.

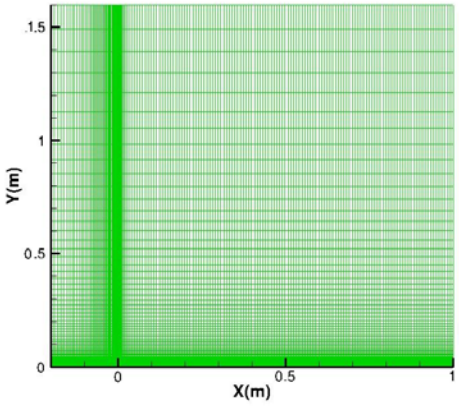

Figure 4.4: Flat plate grid.

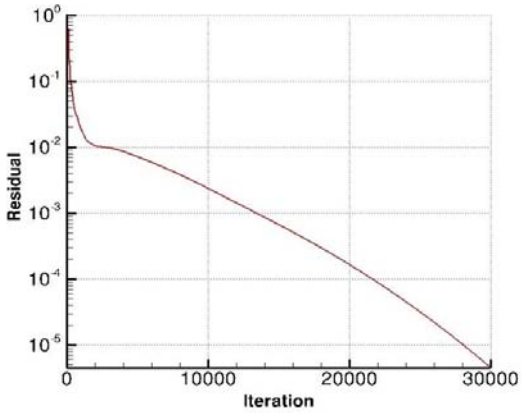

Figure 4.5: Residual convergence, flat plate test case.

The skin friction results with the S-A model and the ONERA corrections have been compared in Figure 4.6 with the Mills and Hang [28] semi-empirical relation for turbulent skin friction coefficient over a rough flat plate:

$$C_f = (3.476 + 0.707ln\frac{x}{k_s})^{-2.46} \quad (4.13)$$

The error on the skin-friction coefficient between both models at a Reynolds number of 5 million is within 9%.

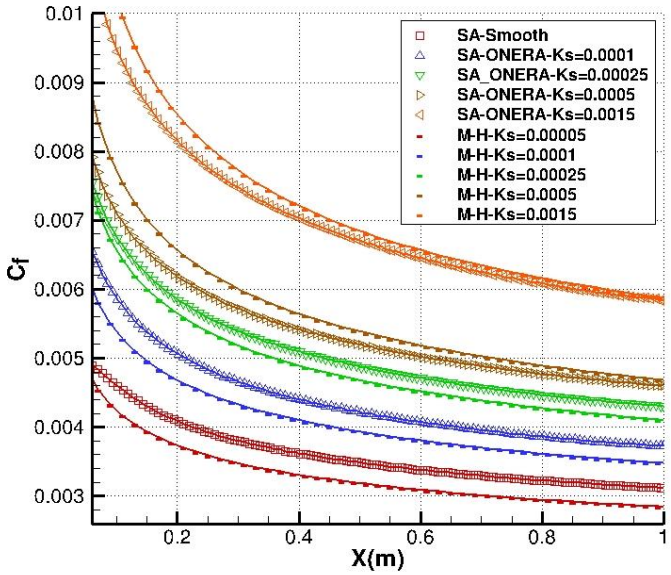

Figure 4.6: Turbulent flat plate skin friction comparison: NSMB (SA-ONERA) with semi-empirical relation.

4.3.1.2 CL-α over NACA0012 airfoil (smooth and rough)

Computations were carried on a clean NACA0012 airfoil geometry with smooth and rough walls with a Reynolds of 6 million, Mach 0.2 using the SA-ONERA turbulence model. The wall roughness value is 0.2794 mm [103]. A 3D extruded O-mesh with 60,000 nodes with minimum normal wall spacing of 10^{-6} chord ($y^+<1$) with an expansion ratio of 1.5 was generated by extruding a 2D O-Mesh, since NSMB is a 3D solver. The airfoil has 299 points on the surface, with concentrations at the leading and trailing edge regions as shown in Figure 4.7. The far-field distance was 100 chord to ensure proper lift and drag values. Convergence levels reached 10^{-5} on the L2-norm of the density residual, as seen on Figure 4.8. The C_L-α curve for the smooth and rough NACA0012 are shown in Figure 4.9. Lift values at constant angle of attack are differing by less than 5% up to 11 degrees before stall. C_{Lmax} is predicted within 7% and 5% for the clean and rough cases, respectively.

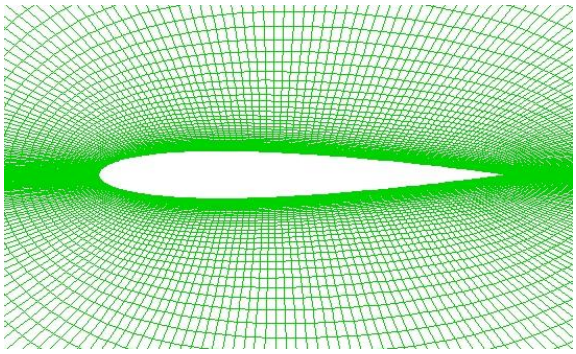

Figure 4.7: NACA0012 N-S grids.

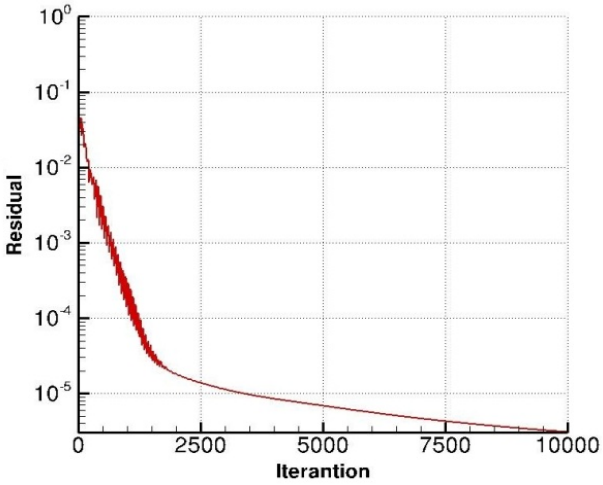

Figure 4.8: Residual Convergence, NACA0012 test case.

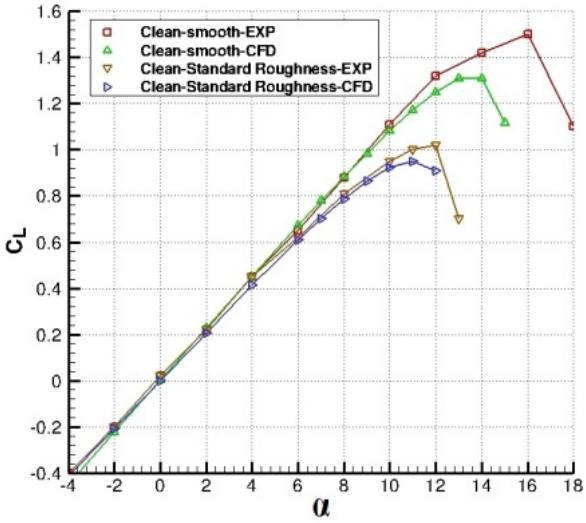

Figure 4.9: CL-Alpha graph for NACA0012 smooth and rough surface.

4.3.1.3 2D transonic RAE2822 airfoil

For completeness of the 2D validation, a transonic test case is used in our validation [104]. Experimental results from the AGARD test case 6 of the RAE2822 smooth airfoil, namely 2.31° angle-of-attack, Mach 0.729, Reynolds 6.5 million were compared to numerical results using the Spalart-Allmaras turbulence model [39]. The 3D C-mesh was generated by extruding a 2D C-mesh with 350 grid points in the wake-cut/airfoil and 50 in the normal directions totaling 35,000 nodes with minimum wall spacing of 2 10^{-6} chord ($y^{+}<1$) and expansion ratio of 1.5 as shown in Figure 4.10. The far-field distance was located 100 chords from the mean aerodynamic center. The solution density residual convergence is shown in Figure 4.11, achieving 7 orders reduction. Comparison of the pressure coefficients with results of the WIND and NPARC[1] CFD codes and the experimental data is shown in Figure 4.12, confirming the accuracy of the NSMB solver.

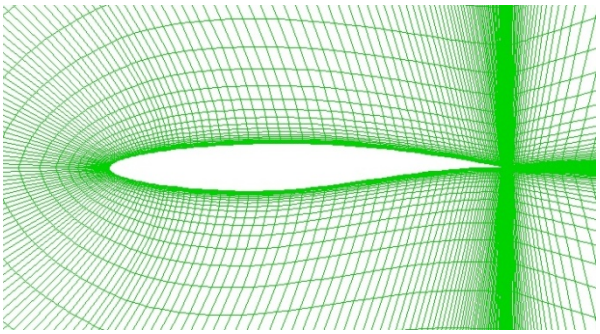

Figure 4.10: RAE2822 N-S grids.

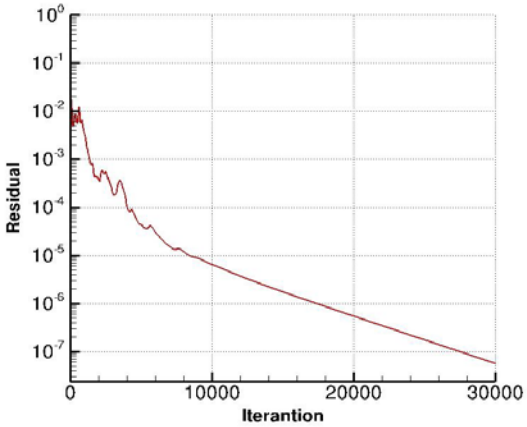

Figure 4.11: Residual convergence, RAE2822 test case.

[1] NPARC Alliance Verification and Validation Archive Web Site

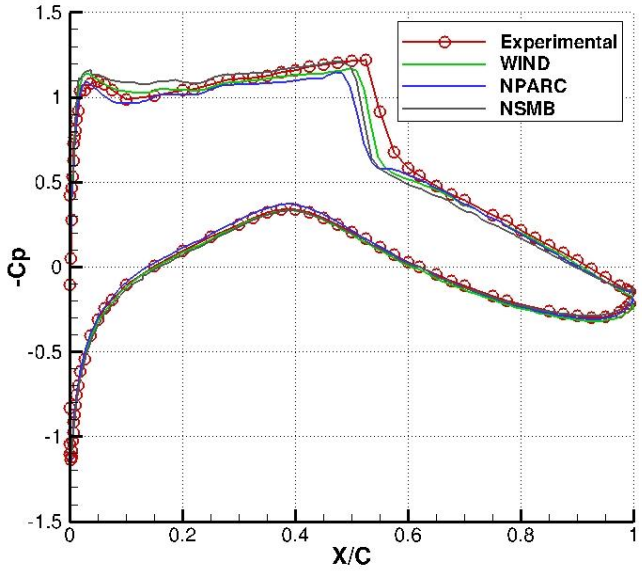

Figure 4.12: Pressure coefficients comparison for the case RAE2822.

4.3.2 Roughness effects on multi-time steps ice accretion

The CANICE2D-NS code was used to perform multi-step ice growth simulation with uniform roughness modeling. The Spalart-Allmaras turbulence model with the ONERA rough wall extension, SA-ONERA, was used. The boundary conditions of the discretized airfoil domain include the airfoil solid surface, the symmetry planes and farfield boundaries. Throughout the calculations, the automatic grid generation/re-generation process using the ICEM CFD mesh generator with elliptic smoothing, described in Section 4.2.5, was used. The effects of surface roughness are examined on two different case studies over the NACA0012 airfoil: one rime ice and the other glaze ice.

4.3.2.1 Rime ice NACA0012 run 405

The effect of surface roughness is analyzed with CANICE2D-NS for the NACA0012 rime ice case 405. The study also indirectly validates the underlying roughness model extension of ONERA to the Spalart-Allmaras turbulence model, as well as the multi-step grid re-generation process with ICEM-CFD. The test conditions are listed in Table 4.1 [48].

Table 4.1: NACA0012 run 405 test conditions.

Chord Length	0.5334 m
Angle of Attack	3.5 deg.
Flight Speed	102.8 m/s
Static Temperature	250.37^{o}K
Mach Number	0.3242
Reynolds Number	4.62 10^{6}
LWC	0.55 g/m^{3}
MED	20 μm
Icing Duration	420 sec.
CANICE2D-NS Icing Time-Steps	4 times 105-sec. steps

The clean airfoil droplet impingement efficiency graph, commonly referred to as the collection efficiency curve, is shown in Figure 4.13 for the LEWICE, CANICE2D-panel method and CANICE2D-NS icing codes. The selected normalized surface roughness values for analyses, hold constant during each of the calculations, are 0.0001, 0.0005, 0.001, 0.005. The accreted ice shapes for 4 multi-time steps are shown in Figure 4.14. Note that the number of time steps, varying between 4 and 7 intervals, is typically used in previously published panel and Navier-Stokes icing simulations [10, 37]. Comparison of the final roughness based ice shapes with the experimental ice shape is shown in Figure 4.15. The comparison shows that a surface roughness of value 0.005 results in the best match to the experimental ice shape.

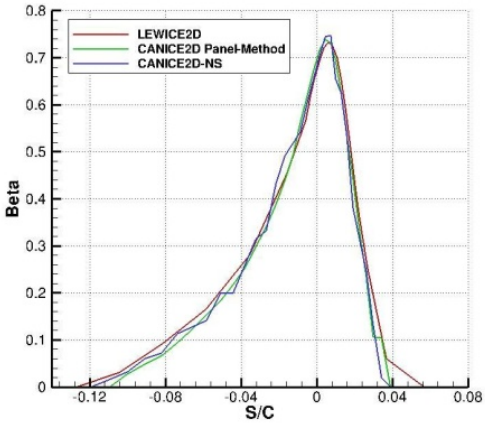

Figure 4.13: Collection efficiency comparison (NACA0012 run 405).

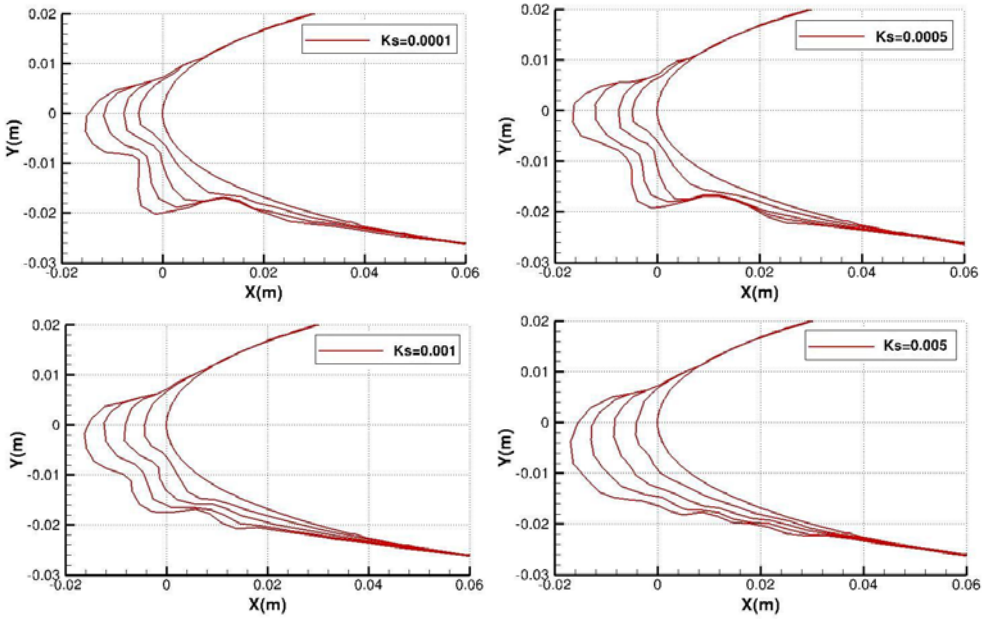

Figure 4.14: Effect of increased roughness on ice shape using CANICE2D-NS (run 405).

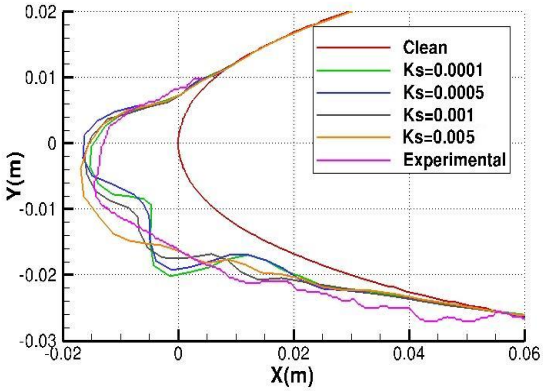

Figure 4.15: Ice shape comparison for different roughness, CANICE2D-NS (run 405).

Comparison of the CANICE2D-NS multi-time step ice shape with the shape obtained by LEWICE and CANICE2D panel-method are shown in Figure 4.16 along with the experimental results [48]. Note that the calculations with LEWICE and CANICE2D use 7 and 4 time steps, respectively. The impingement limits are well captured, as well as the ice thickness distribution. The 3D (2D extruded) structured grid generated with ICEM-CFD, includes around 40,000 nodes and wall spacing of 2 10^{-6} chord (y^{+}<1) on the first layer on each plane. The expansion ratio is 1.5 up to a far field distance located 100 chord away from the airfoil. The computational mesh generated after the 4th time step is shown in Figure 4.17 and shows that the quality of the mesh has been retained throughout the procedure. The convergence of the residual using CANCIE2D-NS with normalized roughness value of 0.005 for this case is shown in Figure 4.18. The density residuals converge to more than 4 orders at each time step intervals.

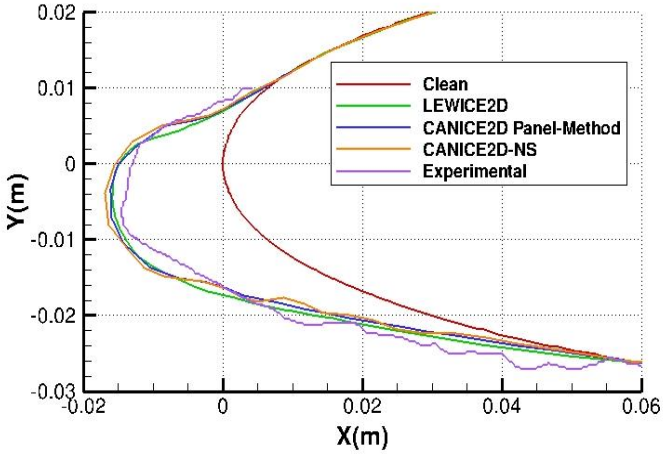

Figure 4.16: Ice shape comparison (NACA0012 run 405).

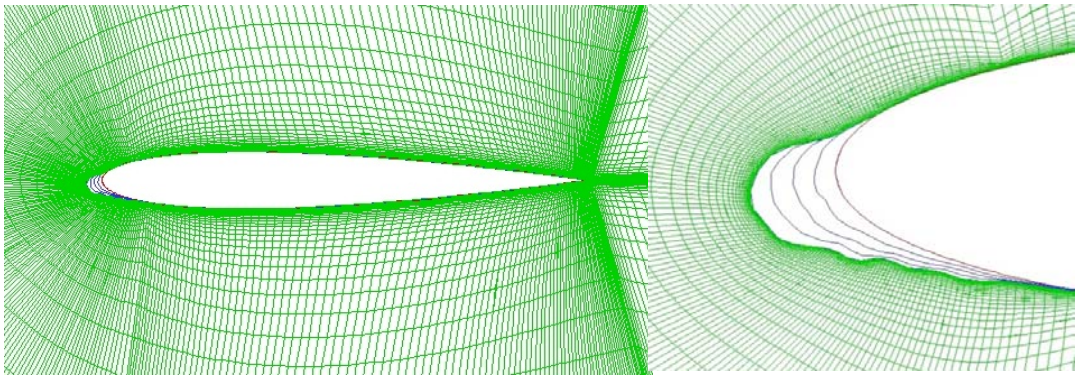

Figure 4.17: NACA0012 run 405: C-mesh using automated ICEM grid generation for CANICE2D-NS multi-time step run.

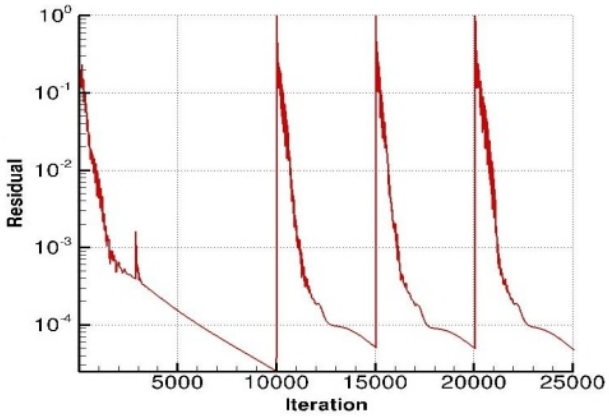

Figure 4.18: NSMB convergence for 4 time-steps of CANICE2D-NS (case 405).

4.3.2.2 Glaze ice NACA0012 run 408

The effect of surface roughness is now analyzed with CANICE2D-NS for the NACA0012 rime ice case 408 with the Spalart-Allmaras turbulence model with the ONERA rough wall extension, SA-ONERA as well as the ICEM-CFD mesh regeneration procedure. The test case conditions are listed in Table 4.2 [48]. The clean airfoil droplet impingement efficiency is shown in Figure 4.19, and compares results of CANICE2D-NS with the LEWICE and CANICE2D panel method. The selected normalized surface roughness values for this case, which were hold constant in space and time, are 0.0001, 0.0005 and 0.001. The accreted ice shapes throughout the 5 time steps calculation for each roughness value are shown in Figure 4.20. Comparisons of the final ice shapes for each roughness value with the experimental ice are shown in Figure 4.21. The effect of changes in roughness values affects the ice horn growing directions with the value of 0.0005 providing the best match to the experimental results, compared with the calculated uniform roughness value 0.001 for CANICE2D-panel method using an empirical model [2]. The roughness value for LEWICE2D is less relevant because it used a non-uniform water bead height model.

Table 4.2: NACA0012 run 408 test conditions.

Chord Length	0.5334 m
Angle of Attack	3.5 deg.
Flight Speed	102.8 m/s
Static Temperature	262.04^{o}K
Mach Number	0.3169
Reynolds Number	4.44 10^{6}
LWC	0.86 g/m^{3}
MED	20 μm
Icing Duration	270 sec.
CANICE2D-NS Icing Time-Steps	5 times 54-sec. steps

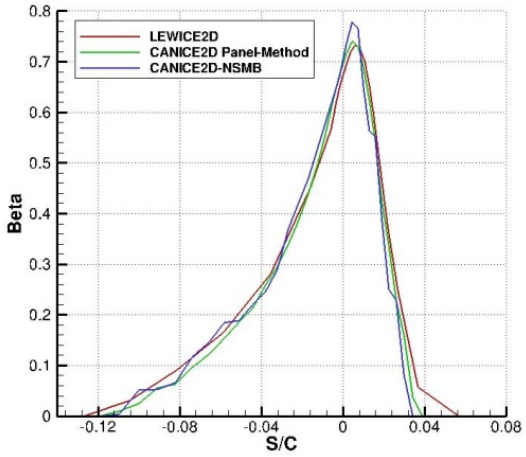

Figure 4.19: Collection efficiency comparison (NACA0012 run 408).

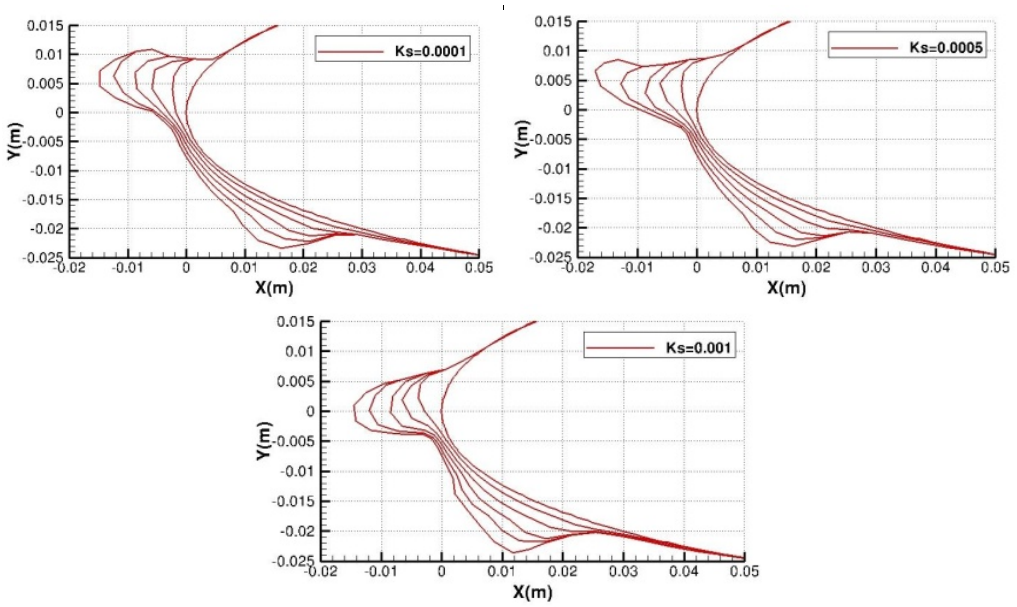

Figure 4.20: Effect of increase in roughness on ice shape using CANICE2D-NS (NACA0012 run 408).

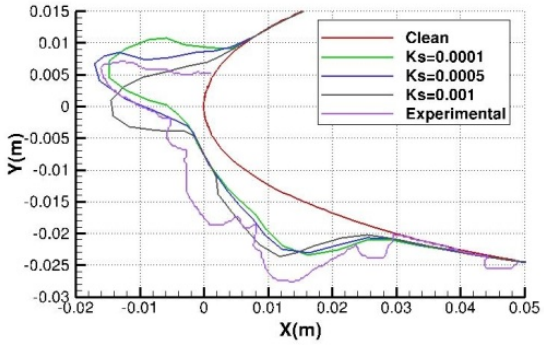

Figure 4.21: Ice shape comparison for different roughness value, CANICE2D-NS (NACA0012 run 408).

Figure 4.22 compares the final 5 time-steps glaze ice obtained with CANICE2D-NS compared to CANICE2D panel-method, LEWICE and experimental data [48]. The LEWICE and CANICE2D panel-method solutions were also run using 5 time-steps. The ice shape horns heights are under predicted by the CANICE2D panel-method. The CANICE2D-NS 5 time-steps ice shape shows better prediction of the upper ice horn, but still underestimates the lower ice horn. The final structured mesh generated for the 5th ice layer is shown in Figure 4.23. The mesh has the same topological properties than described in case 405 above, and manages to conserve its quality throughout the multi-step procedure. The NSMB3D flow solver density residual convergence for the 5 time-steps is shown in Figure 24. Residual reduction of 4 orders is attained on each sequence.

One observation not apparent from the results presented is that increase in ice shape complexity can generate multiple stagnation points which create difficulties in performing accurate boundary layer prediction. The CANICE-NS code formulation used here removes these difficulties altogether.

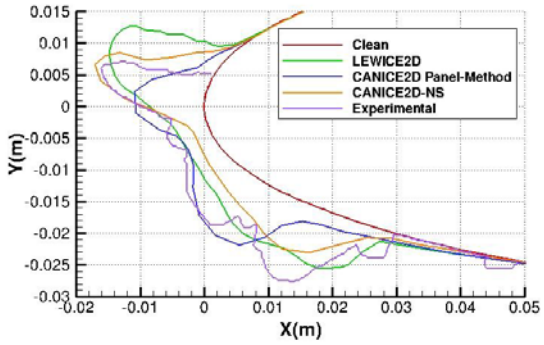

Figure 4.22: Ice shape comparison for NACA0012 run 408.

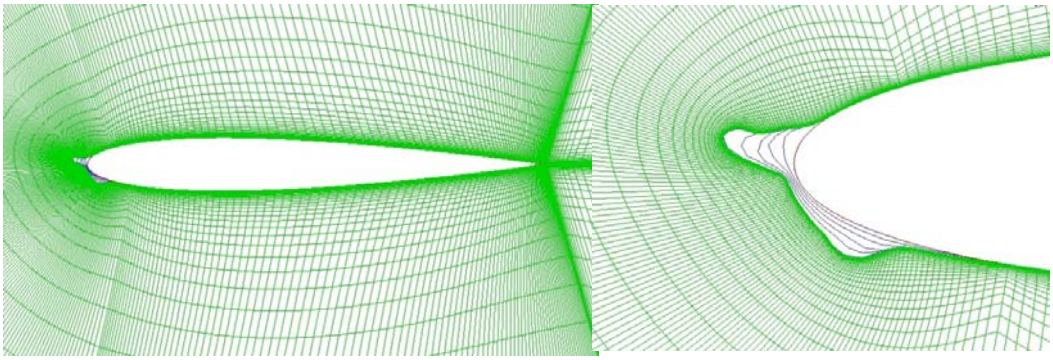

Figure 4.23: NACA0012 run 408 C-mesh using automated ICEM grid generation for CANICE2D-NS multi-time step run.

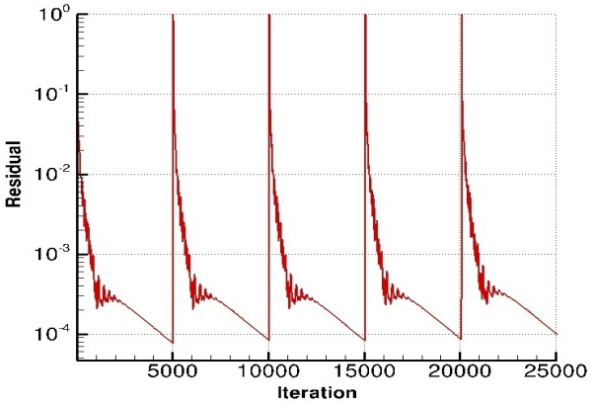

Figure 4.24: NSMB convergence for 5 time-steps of CANICE2D-NS (case 408).

4.3.3 Aerodynamic performance analysis

The aerodynamic performance reduction analysis resulting from the accreted ice on the airfoil for the cases above is presented. Calculations are performed all the way up to and beyond stall angles for the iced and clean airfoils. This allows quantification of the reduction in C_{Lmax}, but as important, the qualification of the post-stall behavior. Indeed, aircraft certification requires controllability conditions in the post-stall regime [105]. Thus, the prediction of whether or not abrupt stall occurs is an important quality of an icing simulation framework. In this section, the Spalart-Allmaras turbulence model is used for clean airfoil simulations, and its ONERA rough wall extension used for iced geometries in all flow simulations.

4.3.3.1 Rime ice NACA0012 run 405

The flow conditions for this test case are listed in Table 4.1. The experimental data of aerodynamic characteristics such as lift, drag or pitching moment for this test case is only available for the clean airfoil geometry. For the iced airfoil, we replace the experimental validation by CFD-based comparisons of the aerodynamic characteristics between the flow around the airfoil with experimental ice shape and the CANICE2D-NS ice shape of Section 4.3.2. To perform the comparison, a grid around the airfoil with the experimental ice shape was created with the grid properties and topology similar to the ones used in Section 4.2.5. The generated grid is shown in Figure 4.25. The computed aerodynamic lift as a function of angle of attack using the NSMB3D solver for the clean airfoil and the predicted ice shape airfoil are shown in Figures 4.26. For the simulated ice and experimental ice calculations, a surface roughness value of 0.005 is used. The clean airfoil stall angle for the numerical results is close to 12^o compared to the experimental data of 13-14^o. The simulation of the experimental ice shape geometry shows a stall angle close to 6^o. The numerical solutions for higher angles of attack contain oscillation in the solution density residual convergence and are therefore not included in the graph. The stall angle simulation with CANICE2D-NS is close to 7-8^o, only 1-2^o degree off the experimental ice shape numerical result. Comparison of pressure coefficients for an angle of attack of 6^o are presented in Figure 4.27, where oscillations in the pressure due to the presence of the ice surface seen in Figure 4.25 can be observed. It can be observed that the pressure coefficient distribution predicted by CANICE2D-NS for the numerical and experimental ice shapes are in close agreement which explains why the numerical flow solutions are in agreement.

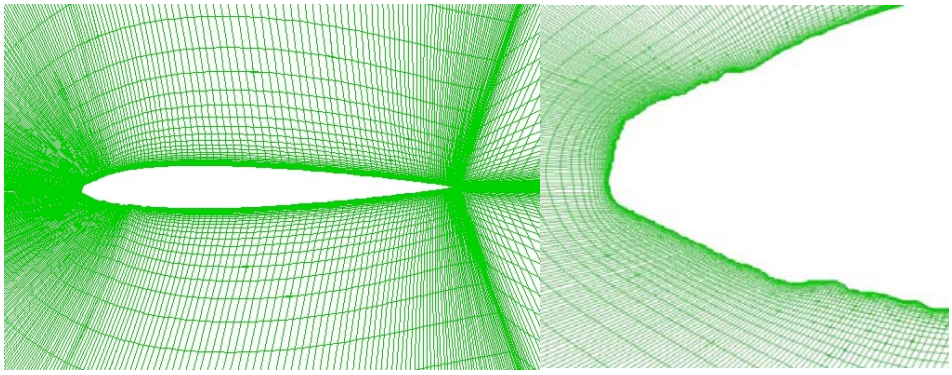

Figure 4.25: ICEM mesh generated experimental ice shape (case 405).

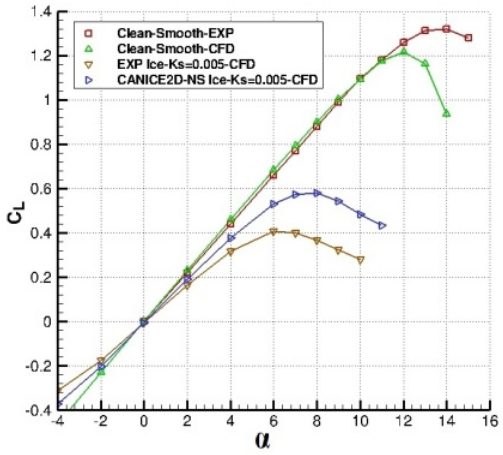

Figure 4.26: CL comparison (case 405).

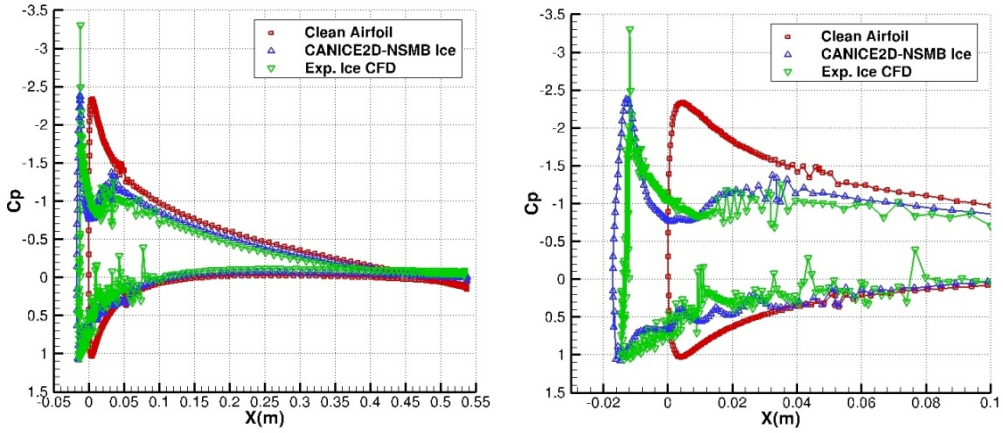

Figure 4.27: Pressure coefficient comparison (α=6^{o}, case 405). (Right, leading edge zoom).

4.3.3.2 Glaze ice NACA0012 run 408

The flow conditions for this test case are listed in Table 4.2. Similar to the previous test case, numerical predictions were performed with the experimental ice shape geometry to perform numerical comparisons between clean and iced geometries. The smoothed grid generated for the

experimental ice shapes of case 408 using the grid generation method of Section 4.2.5 is shown in Figure 4.28. Note that this geometry possesses severe leading edge curvature with sign changes, and that the mesh regeneration process does not fail. The close match in simulated ice shape shown in Figure 4.22, particularly the upper horn position and shape, results in a close match in aerodynamic lift coefficient prediction (see Figure 4.29).

The aerodynamic performance analysis is performed using the automatic process to determine the lift coefficients as function of the airfoil angle of attack. For the CANICE2D-NS simulated and experimental ice shapes, a surface roughness value of 0.0005 was selected for the flow simulation. The results are shown in Figure 4.29. The simulation of the clean airfoil shows a stall angle close to 13^o compared to the experimental stall angle of 13^o. Both experimental ice shape and CANICE2D-NS ice shape CFD simulation shows stall angle close to 6^o. Pressure coefficients comparison for the clean airfoil and the predicted ice shapes for angle of attack of 6^o are shown in Figure 4.30. It can be observed that the pressure distribution for the CANICE2D-NS ice shape is in close agreement with the pressure distribution predicted for the experimental ice shape, thereby providing similar stall angles.

Figure 4.29, also includes the performance of the converged ice shape using CANICE2D-NS with 160 time steps. It shows the stall angle close to 9^o, compared to 5 time steps ice result of 6^o. The 5 time step ice shape is closer to the experimental ice shape which results in better prediction of the stall angle of attack, but the ice shape is not numerically converged. The 160 time steps ice shape shows lower ice accumulation compared to the experimental ice which results in the prediction of the stall point at a higher angle of attack, but the ice shape solution is numerically consistent. This issue will be further discussed in the next section.

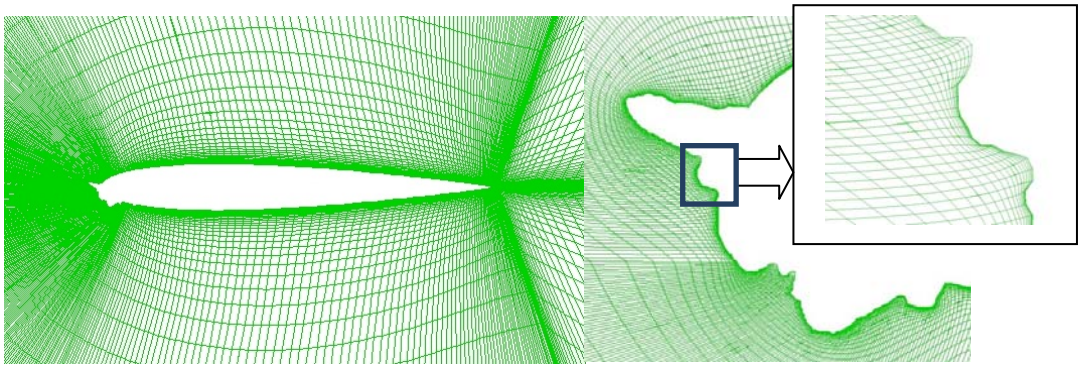

Figure 4.28: ICEM mesh generated for experimental ice shape (case 408).

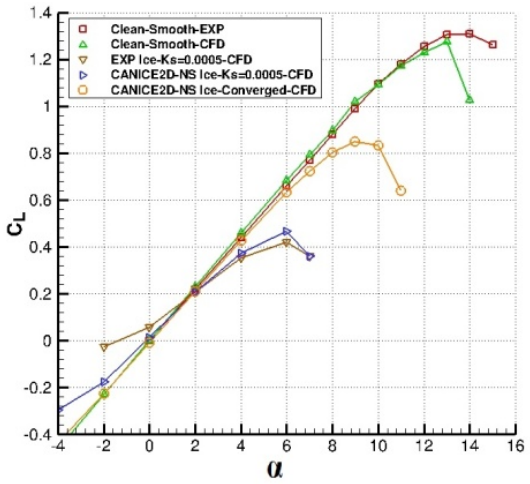

Figure 4.29: CL comparison (case 408).

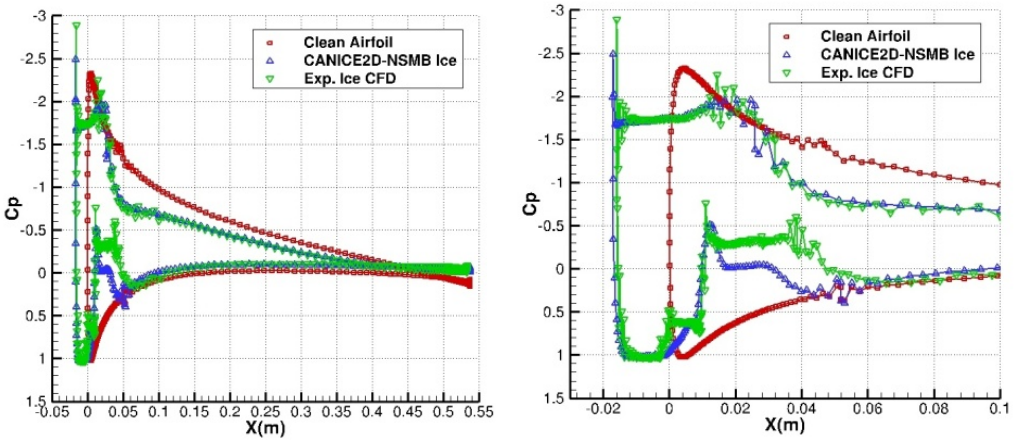

Figure 4.30: Pressure coefficient comparison ($\alpha=6^{o}$, case 408). (Right, leading edge zoom).

4.3.4 Convergence analysis of the numerical solution of multi-time step ice accretion prediction

The objective of this section is to achieve time-independent numerical results of the quasi-steady ice accretion simulation. The number of time steps has been parameterized and varied from 4 to 160 to verify the convergence of the numerical scheme used in the CANICE2D-NS framework. The multi-step procedure is compared to results obtained with the CANICE-panel method based solution framework.

4.3.4.1 Rime ice NACA0012 run 405

The numbers of time steps used for this test case were 4, 8, 16, 32, 64 and 128 steps, with results shown in Figure 4.31. The ice shapes simulated by CANICE2D-NS converge asymptotically towards the result obtained with 128 steps. It can be inferred that, within

engineering accuracy, the results using 32 and 64 layers are converged. Although the solution of the CANICE2D panel method appears reasonable, the numerical results fail to converge when increasing the number of time steps, as shown in Figure 4.32.

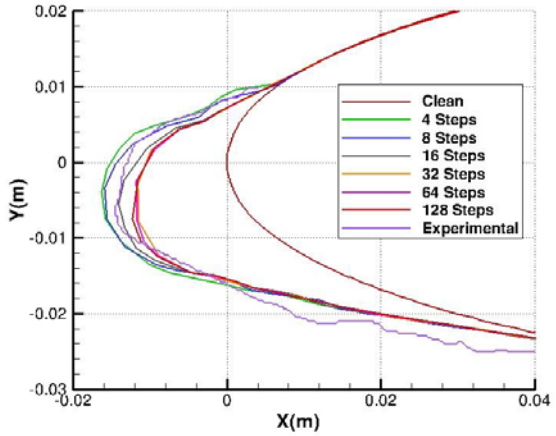

Figure 4.31: NACA0012 run 405 ice shape comparison with increasing time steps (CANICE2D-NS).

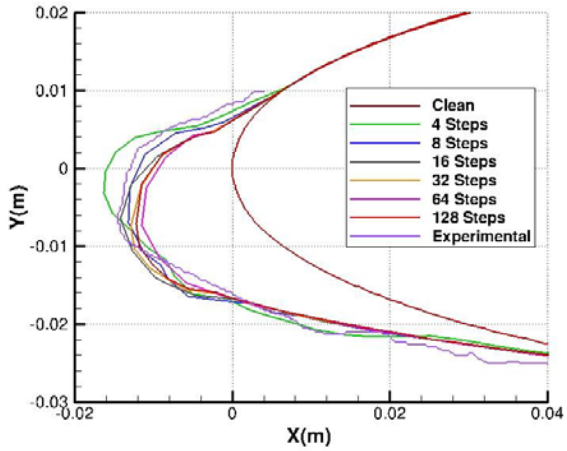

Figure 4.32: NACA0012 run 405 ice shape comparison with increasing time steps (CANICE2D-Panel Method).

4.3.4.2 Glaze ice NACA0012 run 408

For this test case, the number of time step chosen for the calculations were 5, 10, 20, 40, 80 and 160 time steps, with results shown in Figure 4.33. The choice stems from the fact that most icing simulations for this test case are performed with 5 steps, which were doubled until 160 time steps. Once again, the results for the CANICE-NS framework are converging asymptotically towards the finest time step results. Results obtained with 40 time steps are reaching engineering

accuracy. However, the solution of the CANICE2D panel method fails to converge when increasing the number of time steps, as seen in Figure 4.34.

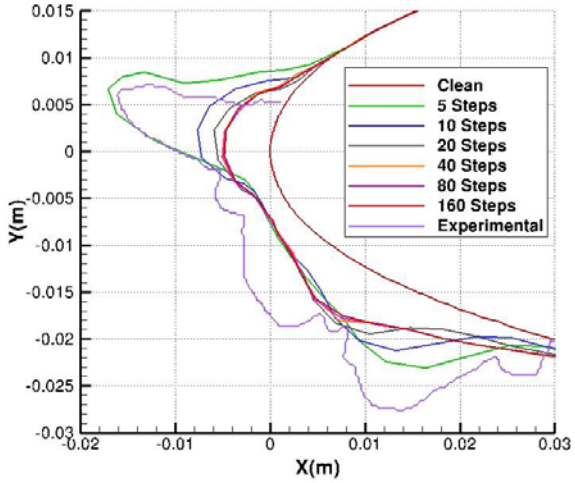

Figure 4.33: NACA0012 run 408 ice shape comparison with increasing time steps (CANICE2D-NS).

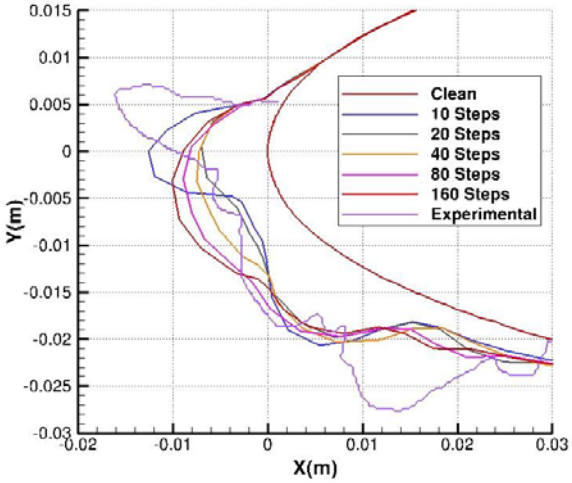

Figure 4.34: NACA0012 run 408 ice shape comparison with increasing time steps (CANICE2D-Panel Method).

4.3.4.3 Multi-time steps results analysis

Figure 4.35 shows the convergence of the ice shape results using a Root Mean Square of the ice shape difference between the multi-step and the reference multi-step ice shapes, the reference profile being the one with the maximum number of time steps. The figure shows that the method converges asymptotically. However, the results of Section 4.3.4 show that the converged ice shapes are agreeing less with the experimental ice shapes than the shapes obtained running the simulations on a small number of time step intervals. This is due to the fact that the combination

of small number of time steps intervals with selection of appropriate roughness values were previously tuned to match the experimental results, but without satisfying convergence criteria. Time-independent results need to have an order increase in time steps, around 40-60 compared to 4-5 used previously.

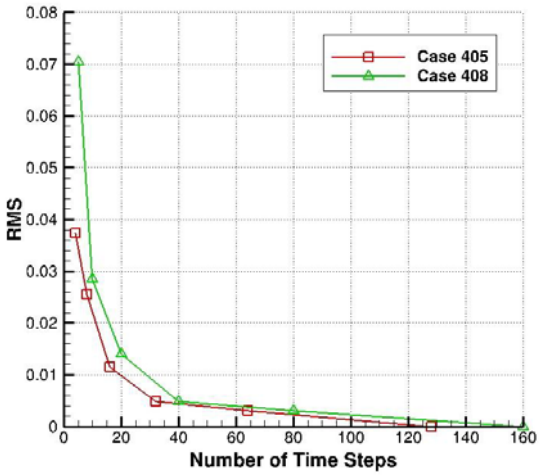

Figure 4.35: RMS comparison for the multi-time steps ice shape convergence (cases 405 and 408).

4.4 Conclusion

In this paper, simulation results of iced airfoil geometries and aerodynamic performance using a newly developed Reynolds Averaged Navier-Stokes based two-dimensional ice accretion framework CANICE2D-NS are presented. The code is based on the CANICE2D-panel method framework and has been augmented with the addition of the NSMB Multi-Block Navier-Stokes solver. The skin-friction coefficients are obtained from the Navier-Stokes solution and used to calculate the heat transfer coefficients through empirical relations. During the process, the grids are obtained by using ICEM CFD with elliptic smoothing using a mesh regeneration approach. To facilitate multi-time step ice accretion simulations, the framework has been entirely automated. It has been shown that the Navier-Stokes solver and grid generation process provides an accurate and robust process for application to complex ice shapes simulations, namely rime and glaze ice accretion. The importance of using rough wall modeling, here using the ONERA rough wall treatment implemented in the Spalart-Allmaras turbulence model is demonstrated. The aerodynamic performance analysis of the accreted ice shapes, specifically the lift-curve, have shown that the simulation leads to predict the stall angle and C_{Lmax} values which were not

possible with the panel-method based framework. A convergence analysis of the multi-step process demonstrated that the CANICE2D-NS code provides asymptotically time-independent results between 40-60 time steps whereas the CANICE2D-Panel method failed to converge. The results are moreover showing that time-independent results are in less agreement with experimental results than when using small number of time steps intervals because the roughness values of the test cases have been previously selected to match non-converged solutions to experimental results. The converged multi-step procedure will require new baselines of roughness values while providing a rigorous approach to further our advancements in reducing modeling errors in the ice accretion process.

CHAPTER 5 ARTICLE 2:

ADAPTIVE CURVATURE CONTROL GRID GENERATION ALGORITHMS FOR COMPLEX GLAZE ICE SHAPES RANS SIMULATION

Article accepted in Journal of AIAA (September 2015), written by:

Kazem Hasanzadeh Lashkajani, Eric Laurendeau, Ion Paraschivoiu

École Polytechnique de Montréal

Abstract

The paper presents the developments of novel mesh generation algorithms over complex glaze ice shapes containing multi-curvature ice accretion geometries, such as single/double ice horns. The twofold approaches tackle surface geometry discretization as well as field mesh generation. First, an adaptive curvilinear curvature control algorithm is constructed solving a 1D elliptic PDE equation with periodic source terms. This method controls the arclength grid spacing so that high convex and concave curvature regions around ice horns are appropriately captured and is shown to effectively treat the grid shock problem. Second, a novel blended method is developed by defining combinations of source terms with 2D elliptic equations. The source terms include two common control functions, Sorenson and Spekreijse, and an additional third source term to improve orthogonality. This blended method is shown to be very effective for improving grid quality metrics for complex glaze ice meshes with RANS resolution. The performance in terms of residual reduction per non-linear iteration of several solution algorithms (Point-Jacobi, Gauss-Seidel, ADI, Point and Line SOR) are discussed within the context of a full Multi-grid operator. Details are given on the various formulations used in the linearization process. It is shown that this performance of the solution algorithm depends on the type of control function used. Finally, the algorithms are validated on standard complex experimental ice shapes, demonstrating the applicability of the methods.

5.1 Introduction

There has been significant progress in the field of aircraft ice prediction based on the Reynolds Averaged Navier-Stokes (RANS) approach. New improvement such as turbulence models with roughness effects, convergence acceleration techniques, parallel computing, unsteady flow calculations, etc. have improved the usability of RANS based approaches in the field of ice accretion simulation and effects analysis [52, 106, 107]. A common difficulty resides in mesh generation around complex glaze ice shapes. There have been many efforts in use of structured, unstructured, and hybrid approaches, but they all encounter different difficulties in terms of mesh quality, algorithm automation, numerical accuracy, etc. [4, 36, 38, 107-109]. The focus of this paper is on structured grid generation approach for complex ice shapes. Simplicity and lower memory usage in grid data structure is one advantage of this approach, but still there remain many difficulties when generating grids with sufficient quality for complex domains. The challenges are in the treatment of discontinuities, sharp angles and highly concave and convex areas present on glaze ice shapes [14, 54, 58, 82, 110, 111].

In general, icing grid generation process involves three basic steps: i) description of the iced surface geometry, ii) grid points distributions on the surface [58] and finally iii) generation of the field mesh points. There are three major classes for grid generation: algebraic methods, partial differential equation (PDE) methods and conformal mapping methods. Within these, the PDE approach is chosen as it is easily amenable to extension to 3D. PDE methods can further be categorized as elliptic, hyperbolic and parabolic methods [21-24].

PDE methods, such as those based on the elliptic Poisson equations, are more complex than algebraic methods but provide means to generate high quality grids by selection of control functions. Starting from a generated algebraic grid as initial solution, the algorithm performs an iterative procedure to generate the desired grid. Elliptic equations have been used extensively for grid generation as they allow control of mesh clustering. Functions have been developed to control mesh spacing, orthogonality and curvature metrics [22].

Several grid generation tools have been developed specifically for iced airfoil effects analysis i.e. SmaggIce [25], Turbo-Grid [78] and parabolic structured and semi-structured grid generator of Thompson (ICEG2D) [58]. There are also other grid generation software such as ICEM [20, 42], GRIDGEN [75], ENGRID [84], which are more general and can be used for multi-block structured grid generation but they still have many limitations when dealing with ice shapes.

SmaggIce [25] performs single and multi-block structured algebraic grid generation with surface smoothing for icing cases. SmaggIce mesh efficiency is highly dependent on user specifications: ice surface preparation and smoothing, grid topology, creation of lines, layers, blocks, sub-blocks, local smoothing and so on. The difficulty in its automation is one of the main problems which reduce the flexibility of such methods for multi-step icing CFD based methodologies.

Turbo-Grid [78] performs single and multi-block structured algebraic grid generation with surface and field smoothing. However, when applied on complex geometries such as experimental glaze ice shapes, the tool also suffers from heavy dependence on user inputs to improve the grid quality.

ICEG2D is based on conformal mapping or parabolic structured and semi-structured grid generation approaches which were specifically developed for icing problem by Thompson et al [58]. The preparation of the iced surface is performed using NURBS curves. The distribution of the boundary grid points on the iced airfoil surface has been improved using weight function based on the surface curvature. In the parabolic method, the grid is constructed layer by layer outward from body and extended through the domain. Grid clustering in concave regions, grid opening in convex regions, outer boundary definition, grid properties close to sharp corners and boundaries, multi-elements geometries grid or multi-blocking are issues limiting the use of structured parabolic methods. The semi-structured grid approach also needs complementary heuristic algorithms such as line deletion/insertion functions to treat grid clustering [79].

This paper proposes algorithms that address some of the issues discussed, in particular the treatment of concave/convex regions to reduce the grid shock problem (related to surface mesh generation) as well as choice of control functions to achieve appropriate grid metrics in the field (related to field mesh generation). The 1D curvilinear curvature based algorithm is presented, followed by the field generation algorithms (parabolic, elliptic). A detailed account of the various control functions is made; Sorenson, Spekreijse and a proposed blended approach. Numerical solution of the governing Poisson equations is then discussed. Section 5.3 presents the results of 1D curvature based algorithm and field generation methods. Efforts have been made to compare the proposed blended approaches to standard algorithms. In Section 5.4, the algorithms are finally applied to 2 standard validation cases, namely the clean NLF0414 airfoil and the experimental glaze ice 623 of the same airfoil, before concluding.

5.2 Methodology

5.2.1 Surface adaptive curvature based grid point distribution algorithm

Here a novel 1D elliptic geometry curvature based point distribution algorithm is described. Considering the 2D transformed Poisson equation, a 1D elliptic transformed equation with curvature based source terms is developed to satisfy the curvature based point distribution on the iced body:

$$r_{\xi\xi} = S \tag{5.1}$$

where r is the geometry point distribution arclength, and S is the source term. A number of source functions such as Sign, Sin., Cos., Spline, damping Cos., damping Atan., damping Spline, mixed functions, etc. have been analyzed. Here a simple efficient periodic function has been selected as the source terms to add to the right hand side (RHS) of the elliptic PDE:

$$S = A \cdot (r + \sin(2\pi r)) \tag{5.2}$$

where A is the amplitude constant parameter defined by the user. This parameter adjusts the global intensity of the surface grid points on the curved domains and can change the stability of the numerical elliptic scheme if chosen very large; it is defined as ($0 \leq A \leq 0.001$) for the case studies presented here.

The step by step procedure of the proposed 1D elliptic curvature based PDE computation is as follows. First, the surface curvature is computed by a finite number of points uniformly distributed on the surface. The curvature value is defined as:

$$\mathrm{Curv}_i = -1/\mathrm{Radius}_i \tag{5.3}$$

where Radius_i is the computed radius of the circle mapped on 3 points (i-1, i, and i+1). For each point (and its arclength) the curvature is defined concave (negative) or convex (positive). Then, a number of points (k=2 to n-1) is selected by the user, depending on the curvature selection criterion, to apply the curvature source term S in a piecewise fashion. Note that k=1 and k=n are the first and last point on the 1D geometry. Second, the source term S is computed for the selected geometry points (k=1 to n):

$$S1_i = A \cdot \mathrm{Sin}\left(\frac{r_i - r_k}{r_k - r_{k-1}} \cdot 2\pi\right) \quad \text{for } (k = 2 \text{ to } n\text{-}1) \text{ and if } (r_{k-1} \leq r_i \leq r_k) \tag{5.4}$$

$$S2_i = A \cdot (\mathrm{Curv}_i / \textstyle\sum_{i=1}^{n} \mathrm{Curv}_i) \cdot (r_i - r_{k-1}) \quad \text{for } (k = 2 \text{ to } n\text{-}1) \text{ and if } \left(r_{k-1} \leq r_i \leq r_{k-1} + 0.5(r_k - r_{k-1})\right) \tag{5.5}$$

$$S3_i = A \cdot (Curv_i / \sum_{i=1}^{n} Curv_i) \cdot (r_i - r_k) \quad \text{for } (k = 2 \text{ to } n\text{-}1) \text{ and if } (r_{k-1} + 0.5(r_k - r_{k-1}) \leq r_i \leq r_k) \tag{5.6}$$

$$S_i = S1_i + S2_i + S3_i \tag{5.7}$$

Note that the source term S includes a periodic Sinus function ($S1_i$) with two additional linear functions ($S2_i$ and $S3_i$). Third, the 1D elliptic PDE equation with the RHS source term is solved for the new curvature based point distribution solution. Source terms are updated (because of embedded r_i) at each iteration of the 1D elliptic solver until a minimum residual is reached. The general descritized form of the proposed 1D elliptic PDE is:

$$r_{i-1} - 2r_i + r_{i+1} = S_i \tag{5.8}$$

The central difference discretized form of the 1D elliptic equation is solved using Point SOR or ADI schemes. The case studies providing verification and parametric study of the proposed methodology are presented in Section 5.3.1.

5.2.2 Field grid generation algorithm

The elliptic and parabolic grid generation approaches are described here including the novel blended elliptic method and the solution schemes.

5.2.2.1 Parabolic method

Conformal mapping approaches such as parabolic mapping methods are one of the grid generation classes suitable for ice grid generation. The parabolic structured grid generation method mainly includes two steps which iteratively generates the grid propagating from body throughout the domain. First step is generation of the algebraic reference grid which includes 2 local orthogonal grid layers on the base layer.

$$X_{i,1}^{j} = X_{i,0}^{j} + \delta_i^j n_{i,0}^j \tag{5.9}$$

$$X_{i,2}^{j} = X_{i,1}^{j} + \delta_i^{j+1} n_{i,1}^j \tag{5.10}$$

Second step is smoothing of the reference grid layers using elliptic Poisson equation including stretching and orthogonality control functions which are briefly shown here in generalized form.

$$g_{22}\left((1+\nu)X_{\xi\xi} + PX_{\xi}\right) + g_{11}\left(X_{\eta\eta} + QX_{\eta}\right) = 0 \tag{5.11}$$

$$g_{11} = x_{\xi}^2 + y_{\xi}^2, \quad g_{22} = x_{\eta}^2 + y_{\eta}^2 \tag{5.12}$$

An additional term $(1+\nu)$ is added to the elliptic smoothing to improve the grid line crossing in concave domain [58]:

$$\nu = \max \sqrt{\frac{\max(g_{11},g_{22})}{g_{22}}} \times f(\theta_\xi) \tag{5.13}$$

$$f(\theta_\xi) = \begin{cases} 1 & , \quad 0 \le \theta_\xi < \pi/2 \\ \sin(\theta_\xi) & , \quad \pi/2 \le \theta_\xi < \pi \\ 0 & , \quad \pi \le \theta_\xi \end{cases} \tag{5.14}$$

$$\theta_\xi = \operatorname{atan}((y_{i+1} - y_{i-1})/(x_{i+1} - x_{i-1})) \tag{5.15}$$

After smoothing the reference grid layers, the middle grid layer is used to generate the next parabolic grid layers. This procedure is repeated until the grid reaches the domain outer boundaries. Verification of the method is presented in Section 5.3.2.

5.2.2.2 Elliptic method

Elliptic equations are used extensively for grid smoothing. Control functions such as mesh spacing, orthogonality, and curvature control have been used to increase the grid quality [55, 82, 83, 112]. A new choice of control function for the elliptic grid generation system is presented.

Considering Cartesian coordinates (x,y) for physical space, and curvilinear (ξ,η) for computational space ($0 \le \xi \le \xi_{max}$ and $0 \le \eta \le \eta_{max}$), a simple one-to-one mapping can be done from physical domain to computational domain and reverse (Figure 5.1).

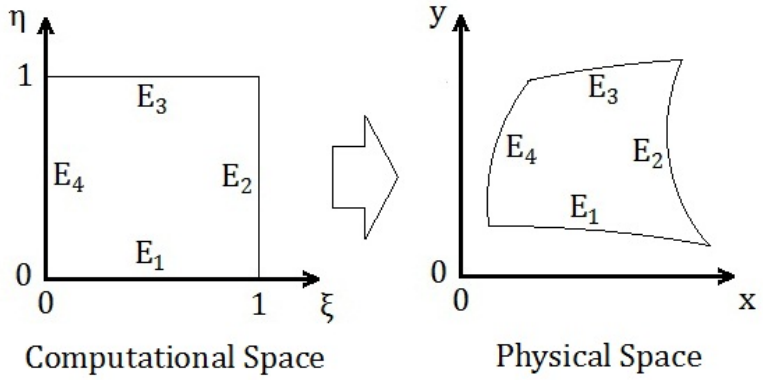

Figure 5.1: Simple mapping: Computational space (left), Physical space (right).

Considering ξ=ξ(x,y) and η=η(x,y), one need to find the mapping equations to satisfy the Poisson equations:

$$\xi_{xx} + \xi_{yy} = P \tag{5.16}$$

$$\eta_{xx} + \eta_{yy} = Q \tag{5.17}$$

The terms P and Q are the forcing functions (control functions). Different choices for these values result in different grid characteristics. Using transformation equations (5.18), the Poisson equations can be written as [82]:

$$\xi_x = y_\eta/J, \quad \xi_y = x_\eta/J, \quad \eta_x = y_\xi/J, \quad \eta_y = x_\xi/J \tag{5.18}$$

$$J = x_\xi y_\eta - y_\xi x_\eta \tag{5.19}$$

$$(x_\eta{}^2 + y_\eta{}^2)x_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta)x_{\xi\eta} + (x_\xi{}^2 + y_\xi{}^2)x_{\eta\eta} = -(x_\xi y_\eta - y_\xi x_\eta)^2(Px_\xi + Qx_\eta) \tag{5.20}$$

$$(x_\eta{}^2 + y_\eta{}^2)y_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta)y_{\xi\eta} + (x_\xi{}^2 + y_\xi{}^2)y_{\eta\eta} = -(x_\xi y_\eta - y_\xi x_\eta)^2(Py_\xi + Qy_\eta) \tag{5.21}$$

The transformed equation is a quasi-linear type equation, which needs to be linearized in order to be solved efficiently.

The choices for control functions here are based on Sorenson (RLS), Spekreijse (SPS), and the developed blended (RLS-SPS-Para) method. The control functions equations are detailed below.

In the Sorenson approach (RLS) [55], P_{RLS} and Q_{RLS} are defined as:

$$P_{RLS}(\xi, \eta) = p(\xi) \cdot e^{-\alpha\eta} + r(\xi) \cdot e^{-c(\eta_{max}-\eta)} \tag{5.22}$$

$$Q_{RLS}(\xi, \eta) = q(\xi) \cdot e^{-b\eta} + s(\xi) \cdot e^{-d(\eta_{max}-\eta)} \tag{5.23}$$

where α, b, c, and d are positive constants defining the propagation of sources on boundaries through domain.

Two main grid controls, grid spacing and grid orthogonality, have been considered. The minimum spacing close to wall along ξ = constant is defined by the user as Δs to control the grid spacing distribution. Also the grid angle θ of layer η = 0 (inner boundary) is defined by the user to control orthogonality of the grid on the body and the propagation through domain. Based on the Sorenson equations, the effect of the terms r and s should be negligible on the layer η=0 (inner boundary), and similarly the terms p and q have negligible effect on the layer $\eta = \eta_{max}$ (outer boundary), so that:

On the inner boundary (η = 0):

$$P_{RLS}(\xi, 0) = p(\xi) \text{ and } Q_{RLS}(\xi, 0) = q(\xi)$$

Using the above correlations and equations (5.20), (5.21):

$$p(\xi) = \left[\frac{y_\eta\left(-\left(\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}\right)/J^2\right) - x_\eta\left(-\left(\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta}\right)/J^2\right)}{J}\right] \tag{5.24}$$

$$q(\xi) = \left[\frac{-y_\xi\left(-\left(\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}\right)/J^2\right) + x_\xi\left(-\left(\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta}\right)/J^2\right)}{J}\right] \tag{5.25}$$

Similarly, on the outer boundary ($\eta = \eta_{max}$), the terms r and s can be computed by the correlation $P_{RLS}(\xi, \eta_{max}) = r(\xi)$, $Q_{RLS}(\xi, \eta_{max}) = s(\xi)$ and equations (5.20), (5.21).

Using the computed derivatives, the terms p, q, r, s are determined and finally sources P_{RLS} and Q_{RLS} are computed through the domain. The iterative process starts by using an algebraic generated grid as the initial solution and the user specified spacing and stretching parameters. The initial values for p, q, r, s are assumed to be zero and input values for Δs and θ are defined. All derivatives of the boundary grid points for the first iteration are computed. The solution of the first iteration is used in the next iteration to solve $x_{\eta\eta}$, $y_{\eta\eta}$ and other derivatives on the boundaries. The values of p, q, r, s for the boundary points are then determined. Now P_{RLS} and Q_{RLS} can be computed for all the interior grid points using Sorenson control function equations. The terms p, q, r, s of the defined control functions can be under-relaxed to increase the numerical stability. This iterative process is continued to satisfy a convergence criterion.

In the Spekreijse approach (SPS) [84], the parametric space as shown in Figure 5.2 is used to compute the source terms as:

$$P_{SPS}(\xi, \eta) = a^{11}p_{11}^1 + 2a^{12}p_{12}^1 + a^{22}p_{22}^1 \tag{5.26}$$

$$Q_{SPS}(\xi, \eta) = a^{11}p_{11}^2 + 2a^{12}p_{12}^2 + a^{22}p_{22}^2 \tag{5.27}$$

$$p_{11} = -T^{-1}\begin{pmatrix} s_{\xi\xi} \\ t_{\xi\xi} \end{pmatrix},\ p_{12} = -T^{-1}\begin{pmatrix} s_{\xi\eta} \\ t_{\xi\eta} \end{pmatrix},\ p_{22} = -T^{-1}\begin{pmatrix} s_{\eta\eta} \\ t_{\eta\eta} \end{pmatrix},\ T = \begin{pmatrix} s_\xi & s_\eta \\ t_\xi & t_\eta \end{pmatrix} \tag{5.28}$$

$$a^{11} = \left(x_\eta{}^2 + y_\eta{}^2\right)/J^2 \qquad a^{22} = \left(x_\xi{}^2 + y_\xi{}^2\right)/J^2 \qquad a^{12} = \left(x_\xi x_\eta + y_\xi y_\eta\right)/J^2 \tag{5.29}$$

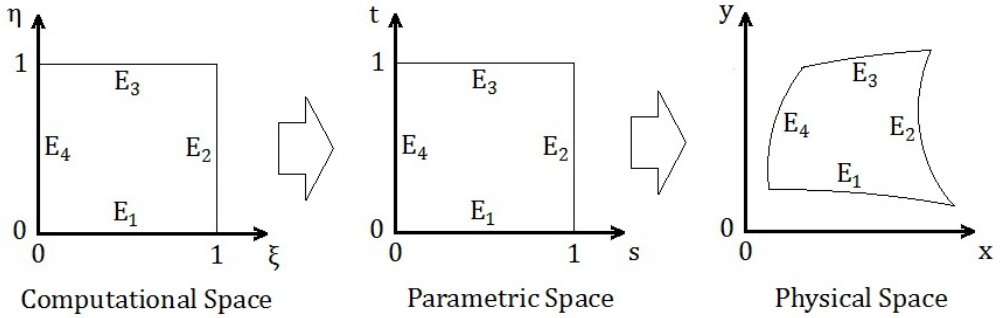

Figure 5.2: Composite mapping: Computational space (left), Parametric space (middle), Physical space (right).

The parametric domain is defined by the nonlinear algebraic transformations. Using the preliminary defined algebraic grid and parametric domain grid, the proposed source terms can be easily computed once for the first iteration and held fixed throughout the entire grid computation.

To include orthogonality, the equations below need to be solved on parametric space using proper boundary conditions to impose the orthogonal boundary of the physical domain:

$$\left(\mathrm{Ja}^{11}\mathrm{s}_{\xi}+\mathrm{Ja}^{12}\mathrm{s}_{\eta}\right)_{\xi}+\left(\mathrm{Ja}^{12}\mathrm{s}_{\xi}+\mathrm{Ja}^{22}\mathrm{s}_{\eta}\right)_{\eta}=0 \tag{5.30}$$

$$\left(\mathrm{Ja}^{11}\mathrm{t}_{\xi}+\mathrm{Ja}^{12}\mathrm{t}_{\eta}\right)_{\xi}+\left(\mathrm{Ja}^{12}\mathrm{t}_{\xi}+\mathrm{Ja}^{22}\mathrm{t}_{\eta}\right)_{\eta}=0 \tag{5.31}$$

These equations have been solved using finite-volume approach with Neumann boundary conditions ($\partial \mathrm{s}/\partial \mathrm{n}=0$) for edges s=0 and s=1 and ($\partial \mathrm{t}/\partial \mathrm{n}=0$) for edges t=0 and t=1. The computed s and t values on boundaries are used to regenerate the parametric grid domain using Hermit interpolation defined by:

$$\mathrm{s}=\mathrm{s}_{\mathrm{E}_1}(\xi)\mathrm{H}_0(\mathrm{t})+\mathrm{s}_{\mathrm{E}_3}(\xi)\mathrm{H}_1(\mathrm{t}) \tag{5.32}$$

$$\mathrm{t}=\mathrm{s}_{\mathrm{E}_2}(\eta)\mathrm{H}_0(\mathrm{t})+\mathrm{s}_{\mathrm{E}_4}(\eta)\mathrm{H}_1(\mathrm{t}) \tag{5.33}$$

$$\mathrm{H}_0(\mathrm{s})=(1+2\mathrm{s})(1-\mathrm{s})^2\ ,\ \mathrm{H}_1(\mathrm{s})=(3-2\mathrm{s})\mathrm{s}^2 \tag{5.34}$$

$$\mathrm{H}_0(\mathrm{t})=(1+2\mathrm{t})(1-\mathrm{t})^2\ ,\ \mathrm{H}_1(\mathrm{t})=(3-2\mathrm{t})\mathrm{t}^2 \tag{5.35}$$

Considering the new generated parametric space (s,t), $\mathrm{P_{RLS}}$ and $\mathrm{Q_{RLS}}$ are computed for the first iteration and applied through the grid computation to impose the condition of orthogonality in the physical space.

In the proposed blended method (RLS-SPS-Para), the different approaches source terms are added with weight factors and decay functions (optional):

$$\mathrm{P}(\xi,\eta)=\mathrm{f_{RLS}}.\mathrm{P_{RLS}}+\mathrm{f_{SPS}}.\mathrm{P_{SPS}}+\mathrm{f_{Para}}.\mathrm{P_{Para}} \tag{5.36}$$

$$\mathrm{Q}(\xi,\eta)=\mathrm{f_{RLS}}.\mathrm{Q_{RLS}}+\mathrm{f_{SPS}}.\mathrm{Q_{SPS}}+\mathrm{f_{Para}}.\mathrm{Q_{Para}} \tag{5.37}$$

in which $\mathrm{f_{RLS}}$, $\mathrm{f_{SPS}}$, and $\mathrm{f_{Para}}$ are the weight factors for Sorenson, Spekreijse (with parametric space generated from algebraic grid normalized arclength), and Spekreijse (with parametric space generated from parabolic grid normalized arclength), respectively. $\mathrm{P_{RLS}}$ and $\mathrm{Q_{RLS}}$ are the Sorenson approach source terms. $\mathrm{P_{SPS}}$ and $\mathrm{Q_{SPS}}$ are the Spekreijse approach source terms where the parametric grid is defined by the generated algebraic grid. $\mathrm{P_{Para}}$ and $\mathrm{Q_{Para}}$ are the Spekreijse approach source terms where the parametric grid is defined by the generated parabolic grid. Optional decay functions such as exponential, linear, or periodic functions are also implemented to control the effects of parabolic grid source terms ($\mathrm{P_{Para}}$ and $\mathrm{Q_{Para}}$) on the elliptic grid

equations. Note that changes in the sources weight factors can result in changes in the stability of the elliptic solution algorithm.

The Spekreijse (SPS) source terms in the blended approach are either

$$P_{SPS} = P_{SPS-SC} \quad \text{or} \quad P_{SPS-SCO} \tag{5.38}$$

P_{SPS-SC} is the Spekreijse source term of spacing and curvature, that uses only the parametric space generated by algebraic grid. $P_{SPS-SCO}$ is the Spekreijse source term of spacing and curvature with additional orthogonality. The orthogonality is achieved by elliptically smoothing the parametric space (with orthogonality boundary condition and algebraic parametric space as the initial solution). Adding Spekreijse orthogonality source term ($P_{SPS-SCO}$) results in a more relaxed problem and a better grid (for smoothed geometries), but a higher computation time because of additional elliptic solution on parametric space and also has difficulty at sharp convex corners. In the remainder of the paper, only the P_{SPS-SC} (named as P_{SPS}) is used for the proposed blended approach.

In the proposed blended approach, the methodology can be summarized as follow. First, generating the algebraic grid and its transformed parametric space. Second, generating the parabolic grid (or the parabolic/algebraic grid) and its transformed parametric space. Third, computing the 2D elliptic transformed Poisson equation with the proposed blended source terms. The initial solution of the iterative solver can be chosen by either the algebraic grid or the parabolic grid. The RLS control functions are computed at each iteration. The SPS and Para control functions are computed only at the first iteration and held fixed throughout the entire grid computation. The decision of the weight factors are based on the two main criteria: the elliptic PDE solution scheme stability and the grid quality criteria decided by the user. The value of each weight factors can be chosen between 0 to 1.

5.2.3 Solution method

5.2.3.1 Linearization

Successive Substitution technique (or Simple Iteration technique) can be used to solve the nonlinear algebraic system. Using the Successive Substitution technique, the transformed Poisson equations (5.20) and (5.21) simplify to:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2\left(Px_{\xi} + Qx_{\eta}\right) \tag{5.39}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2\left(Py_{\xi} + Qy_{\eta}\right) \tag{5.40}$$

$$\alpha = x_\eta{}^2 + y_\eta{}^2 \qquad \gamma = x_\xi{}^2 + y_\xi{}^2 \qquad \beta = x_\xi x_\eta + y_\xi y_\eta \tag{5.41}$$

where the coefficients α, β, γ, J are computed using the last iteration solution, explicitly.

Newton-Raphson technique has also been applied to linearize the problem of Poisson equation. The delta form of equation simplifies to:

$$\left[\frac{I}{\Delta t} - \frac{\partial R}{\partial U}\right] \Delta U = R^k \tag{5.42}$$

where R^k is the computed residual at the time step k. To improve the stability of the iterative scheme, the Jacobian term $(\frac{\partial R}{\partial U})$ is active if $(\frac{\partial R}{\partial U} < 0)$.

5.2.3.2 Solution of system

The linearized transformed Poisson equation by the Successive Substitution technique is solved using Point-Jacobi (explicit), Point Gauss-Seidel (explicit), Point SOR (explicit), Line SOR (implicit) and ADI (implicit). The transformed Poisson equation linearized by the Newton-Raphson technique is solved using Point-Jacobi (implicit) and ADI (implicit) [85]. Note that a Full Multi-grid approach has been applied to reduce the computation time [86, 87, 113].

5.3 Mesh generation results

In this section, different simulation cases are presented with the objective to first verify the implementation. All the grid results are generated with the developed elliptic grid generation package NSGRID. First, a case study for validation of the proposed novel 1D curvature based elliptic PDE surface mesh generation is presented. Then, verification of the correct implementation of the field grid approaches (algebraic, parabolic grid approaches and also the elliptic grid approach with either Sorenson (RLS) or Spekreijse (SPS) control functions) are presented for a clean NACA0012 airfoil. Finally, the proposed novel blended approach is analyzed and validated for field mesh generation of a complex ice shape. The method capability is described and compared with two other approaches using grid quality parameters such as orthogonality, wall spacing, stretching ratio and skewness. The solution methods and convergence rates of the approaches are also discussed. The improvements in grid quality and convergence using the novel blended method are then discussed.

5.3.1 1D elliptic grid curvature based point distribution

The case study is the experimental glaze ice 623 on NLF0414 airfoil. The grid includes 257 points in I and 129 points in J. The experimental ice geometry data is cited in reference [48]. The 1D elliptic curvature based grid point distribution is applied on the surface and the effects of the coefficient A on the spacing along the wall are examined. The curvature computed for the geometry is shown in Figure 5.3. The geometry points with positive curvature higher than 0.4 m^{-1} (green points triangle in Figure 5.3) are selected for the application of the curvature based source terms. It includes 5 surface points with positive high curvature and 2 trailing edge points. The new elliptic point distribution along the geometry is shown in Figure 5.3 for different values of source term coefficient A ($A=2.5\times10^{-4}$, 5×10^{-4}, and 1×10^{-3}). The coefficient A changes the global density of the distributed points solution, which large value of A can cause solution scheme divergence. As shown, the concentration of the point is increased around the 5 selected high curvature regions and the airfoil trailing edge (indicated by the minimum spacing values along the wall). The parabolic grid generated for the geometry point distributions are shown in Figure 5.4, which better show the grid concentrations on and off the body. By increasing the source term coefficient A, the grid concentration on convex domains increases and grid density on concave domains decreases.

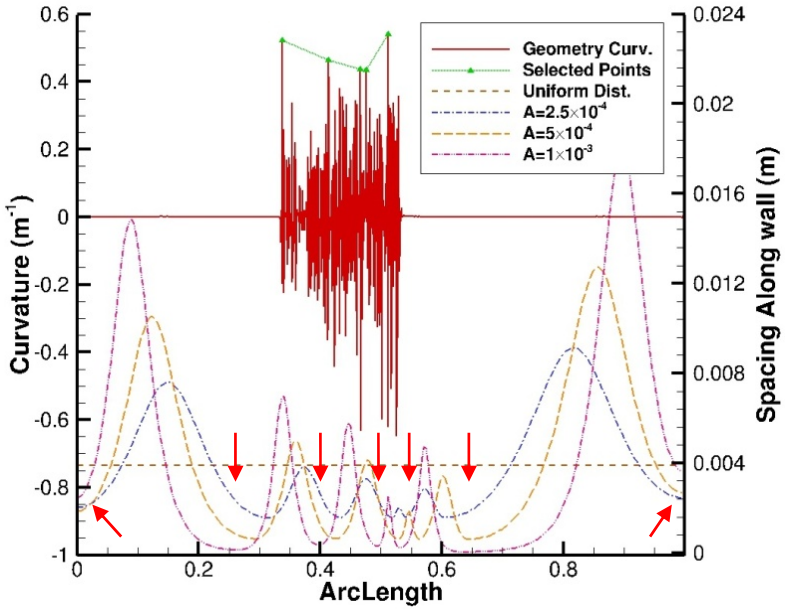

Figure 5.3: Computed curvature (left axis: geometry curvature in red, selected points in green); Curvature based distributed points spacing along the wall (right axis: for different coefficient A).

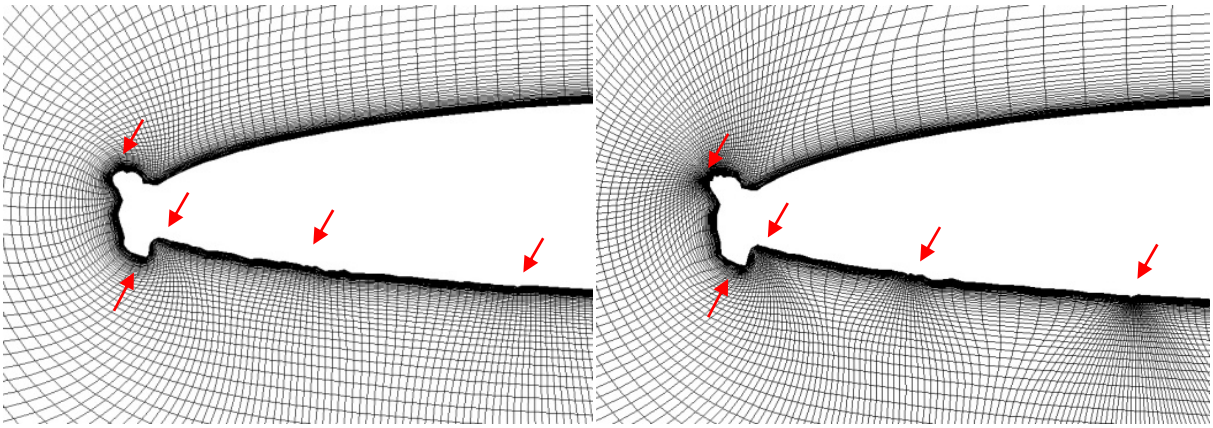

Figure 5.4: Parabolic grid generated for 1D PDE geometry points distribution for two different coefficients A: 2.5×10^{-4}, left, and 1×10^{-3}, right.

The solution convergence of the 1D elliptic point distribution using ADI and Point SOR for different values of A are shown in Figure 5.5. It is seen that Point SOR converges faster than ADI.

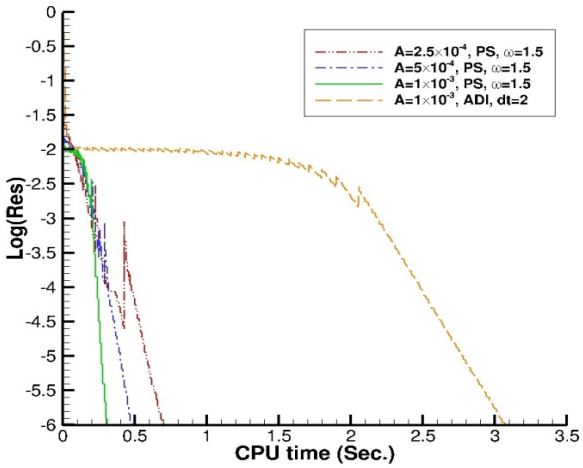

Figure 5.5: 1D PDE Curvature based point distribution for different coefficient A with Point SOR (PS) and ADI.

5.3.2 Field grid generation approaches

5.3.2.1 Clean NACA0012 airfoil

The case study considered here is the simple clean NACA0012 airfoil, since focus is the verification of the correct implementation of the grid generation approaches. The grid includes 129 points in I (tangential) and 65 points in J (normal). All elliptic grid solutions are converged to residual value of 1×10^{-8}. The algebraic and parabolic grids are shown in Figures 5.6 and 5.7, respectively. The elliptic Poisson grid solution with no source terms is shown in Figure 5.8.

These grids are explained by visual inspection. Indeed, the transfinite interpolation (Figure 5.6), the parabolic marching (Figure 5.7) and Laplace operator (Figure 5.8) effects can be easily noticed.

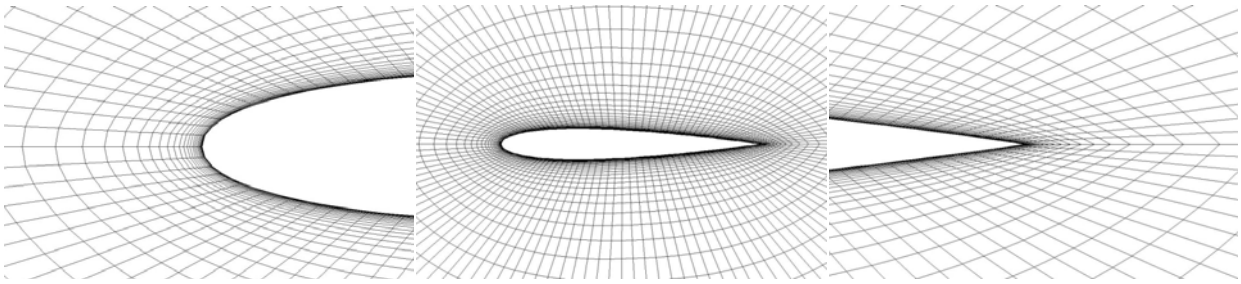

Figure 5.6: Algebraic grid.

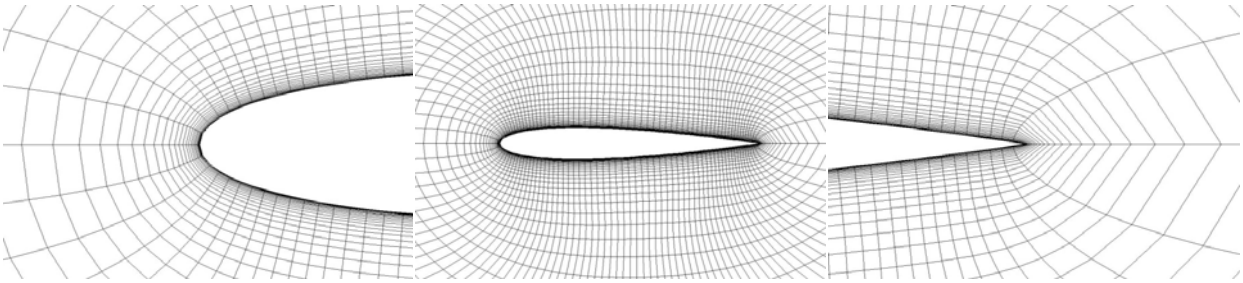

Figure 5.7: Parabolic grid.

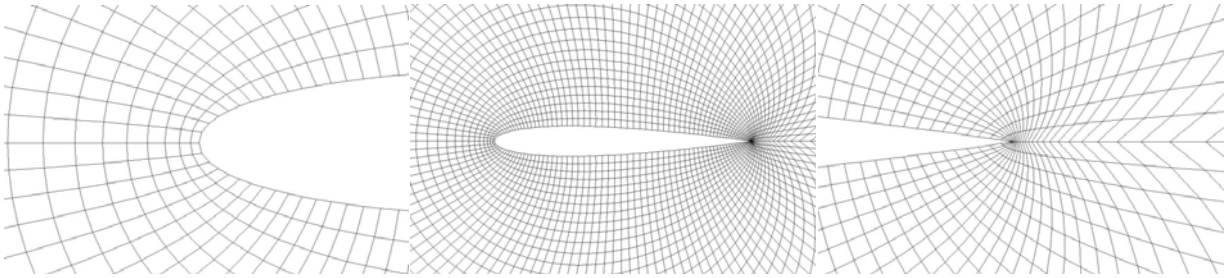

Figure 5.8: Poisson elliptic grid with no source terms.

Finally, the elliptic Poisson grid solution with Sorenson approach (RLS), and Spekreijse approach (SPS) are shown in Figures 5.9 and 5.10, respectively. The effect of Sorenson operator can readily be observed in the mid-chord region (Figure 5.9), where the first larger cell height is significantly reduced. The effect of the Spekreijse operator tends to exacerbate grid skewness near the wall.

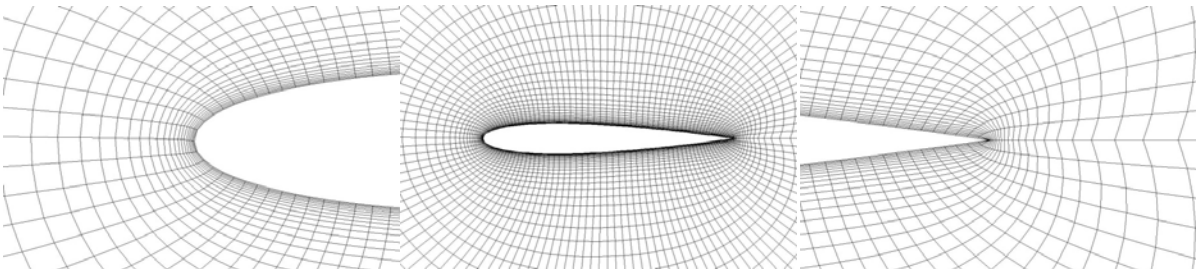

Figure 5.9: Elliptic grid with RLS source terms.

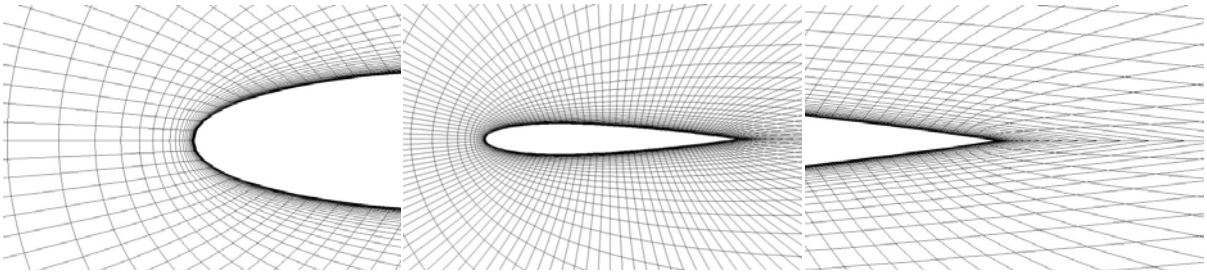

Figure 5.10: Elliptic grid with SPS source terms.

5.3.2.2 Experimental glaze ice case 623-2D on NLF0414 airfoil

The case study is the experimental glaze ice 623 on NLF0414 airfoil. Here focus is on the validation of the proposed blended grid generation approach. The experimental ice geometry data is given in reference [48]. The grid includes 257 points in I and 129 points in J. The 1D elliptic curvature based grid point distribution is applied on the surface with coefficient $A=8\times10^{-4}$ for the surface grid points source term. The curvature computed for the geometry and the new curvature based grid point distribution is shown in Figure 5.11. The geometry points with positive curvature higher than 0.1 m^{-1} (green points triangle in Figure 5.11) are selected for application of the curvature based source terms.

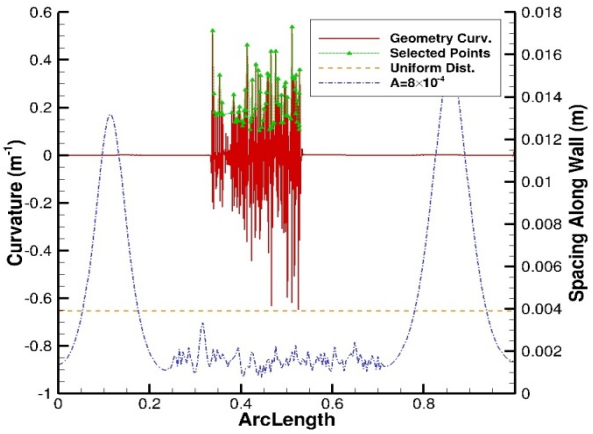

Figure 5.11: Computed curvature (left axis: geometry curvature in red, selected points in green); Curvature based distributed points spacing along the wall (right axis: for coefficient $A=8\times10^{-4}$).

The generated elliptic grids for different source term are shown in Figure 5.12. The first grid is generated using source term RLS with weight factor f_{RLS}= 1. The second grid is generated using blended source terms RLS-SPS with the weight factors of f_{RLS}= 1 and f_{SPS}= 0.5. The third grid is generated using blended source terms RLS-SPS-Para with the weight factors of f_{RLS}= 1, f_{SPS}= 0.5 and f_{Para}= 0.2. A change in weight factors results in different grids. The generated algebraic grid and parabolic grid including their parametric domains are shown in Figure 5.13. The parabolic grid generated in this case includes 80 layers of parabolic grid marching, blended thereafter with the algebraic grid to better control the outer boundaries. The choice of initial solution can be either algebraic or parabolic, but it has been found that the parabolic grid as the initial solution is better conditioned.

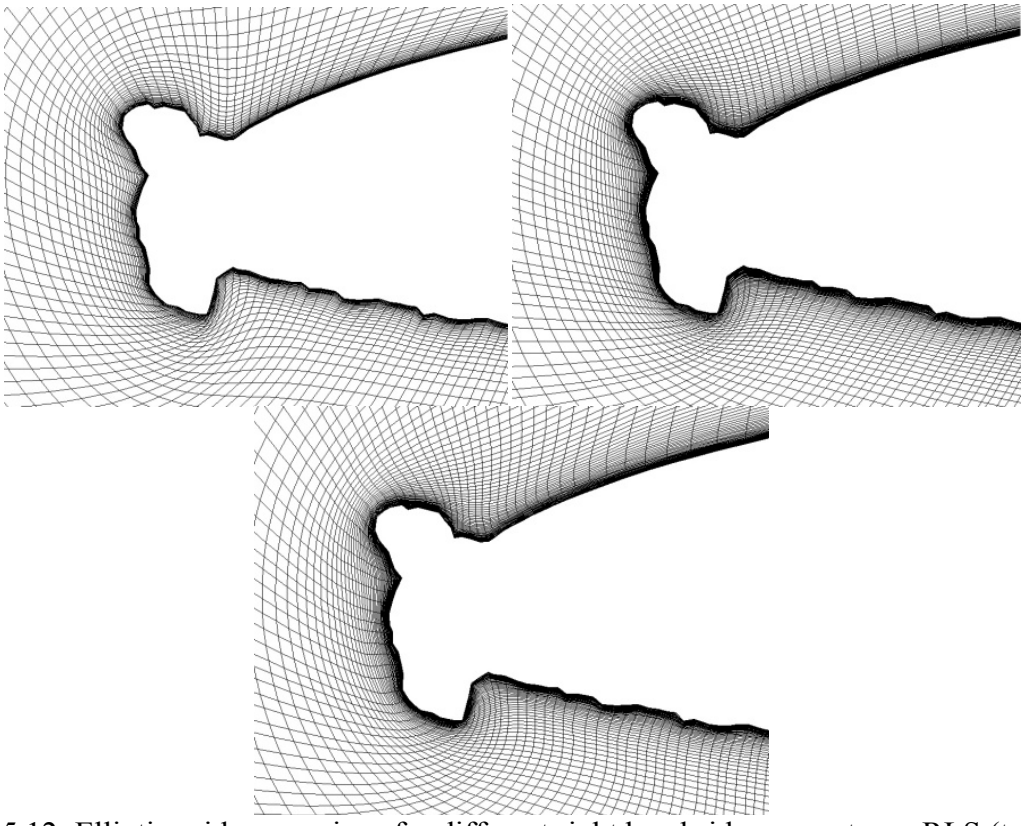

Figure 5.12: Elliptic grid comparison for different right hand side source term: RLS (top left), RLS-SPS (top right), and RLS-SPS-Para (bottom).

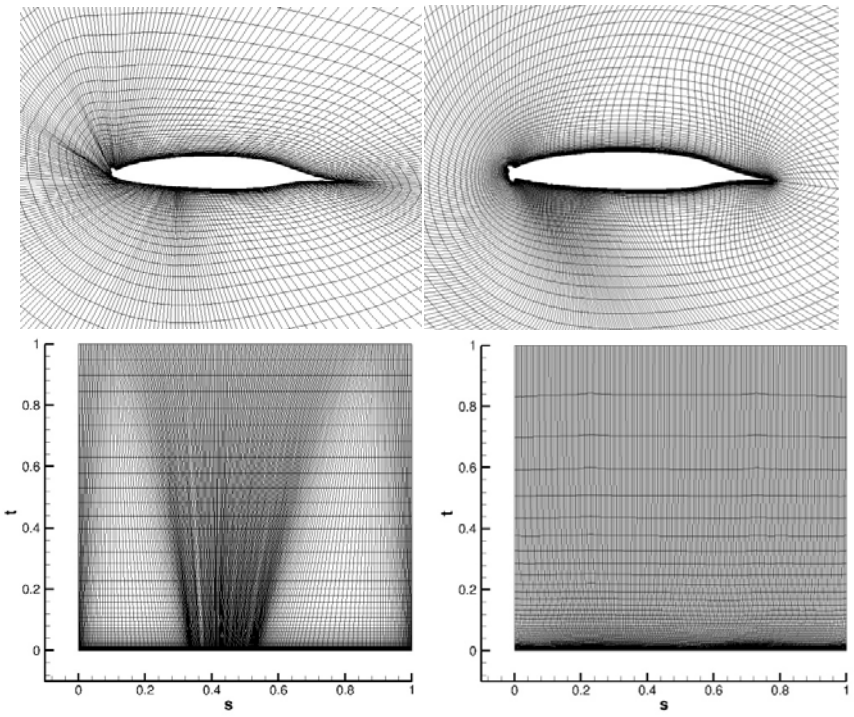

Figure 5.13: Algebraic grid (top left) and its parametric space (bottom left); Parabolic grid (top right) and its parametric space (bottom right).

Blending the source terms on the right hand side of the PDE elliptic equation results in high flexibility on grid parameters control, relaxed problem (or stiffness) and improved convergence properties and computation time. The grid computation time is shown in Figure 5.14, using Line SOR in I (LS-I) with (ω = 1) and 2 levels of Multi-Grid with 20 relaxation on the coarse mesh.

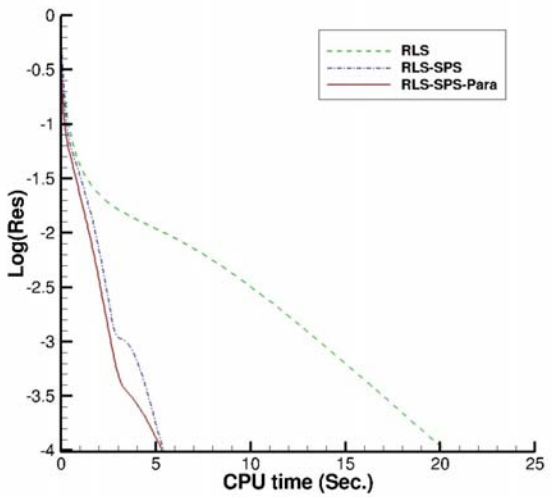

Figure 5.14: NSGRID solution convergence for different blended source terms.

The problem of spacing-curvature control in RLS approach has been improved by adding the SPS source term to the right hand side. The SPS source terms are computed using the parametric space generated by algebraic grid normalized arclength. The elliptic equations are now more relaxed, and the spacing resolution close to wall is improved as shown in Figure 5.15. For more complex geometries, the generated grid still suffers from low orthogonality and negative volumes, which mainly occur in sharp convex corners. An additional source term is added to the right hand side to solve the grid orthogonality issue, maintaining the other grid quality factors proper, such as the minimum spacing resolution, Skewness, stretch ratio, positive volume, etc. The additional source terms are SPS terms computed on the parametric space generated by parabolic grid normalized arclength. It has been observed that the third source term not only improves the orthogonality, but also removes the negative grid volume issues on the surface sharp points that could not be handled by the Sorenson or Spekreijse Source terms.

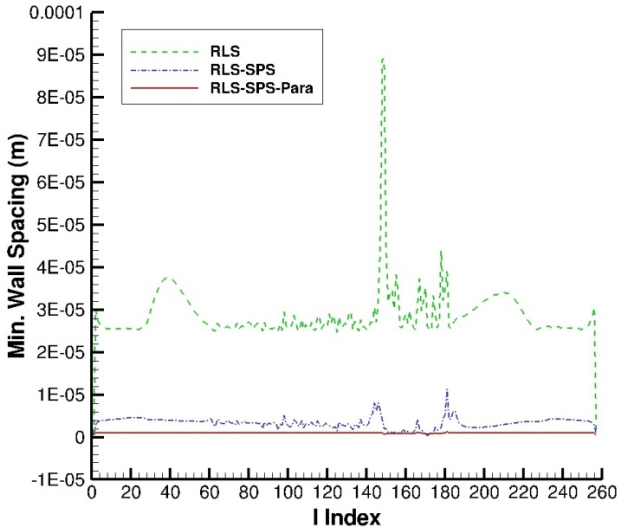

Figure 5.15: Minimum grid spacing close to wall.

Figure 5.16 shows the selected I (148) and J (50) lines to compute the grid quality factors to highlight the generated grids differences. The comparisons of the grid quality factors orthogonalty, skewness and J stretching ratio is shown in Figures 5.17 and 5.18. The comparison of the grid quality factor I stretching ratio is shown in Figure 5.19. Grid quality criterion and their equations are detailed in the Appendix A, Grid Metrics [114]. Orthogonality, skewness and stretching ratio criteria are aimed to be close to 1, 0 and 1, respectively.

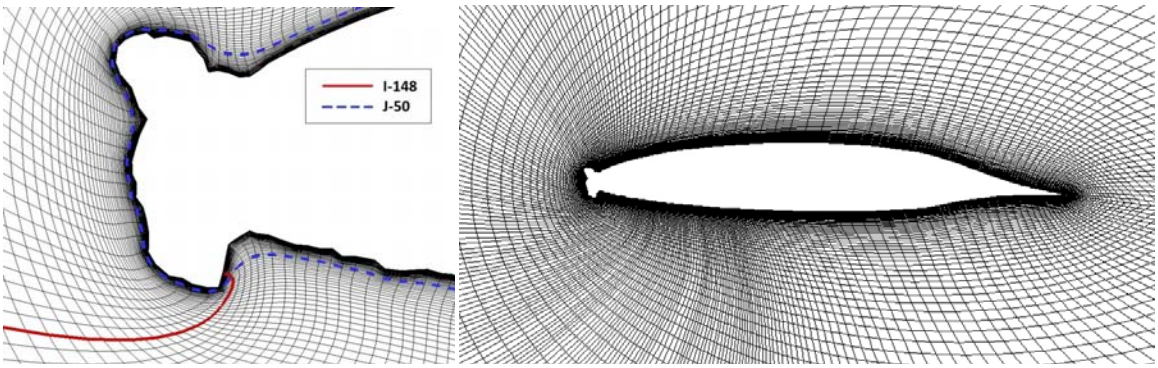

Figure 5.16: Grid selected indexes, I (148) and J (50) to compare the grid quality criteria.

Figure 5.17 compares the grid orthogonality on i=148 for j=1 up to j=50, which properly cover the boundary layer zone of the surface. It shows significant increase in grid orthogonality using RLS-SPS-Para compared to the other source terms. Figure 5.17 also shows the improvement in grid Skewness using RLS-SPS-Para. Figure 5.18 shows proper j stretching ratio using the blended RLS-SPS-Para source terms.

The comparison of I stretch ratio for a selected index j=50 also shows proper range of parameter fluctuation, using RLS-SPS-Para, shown in Figure 5.19.

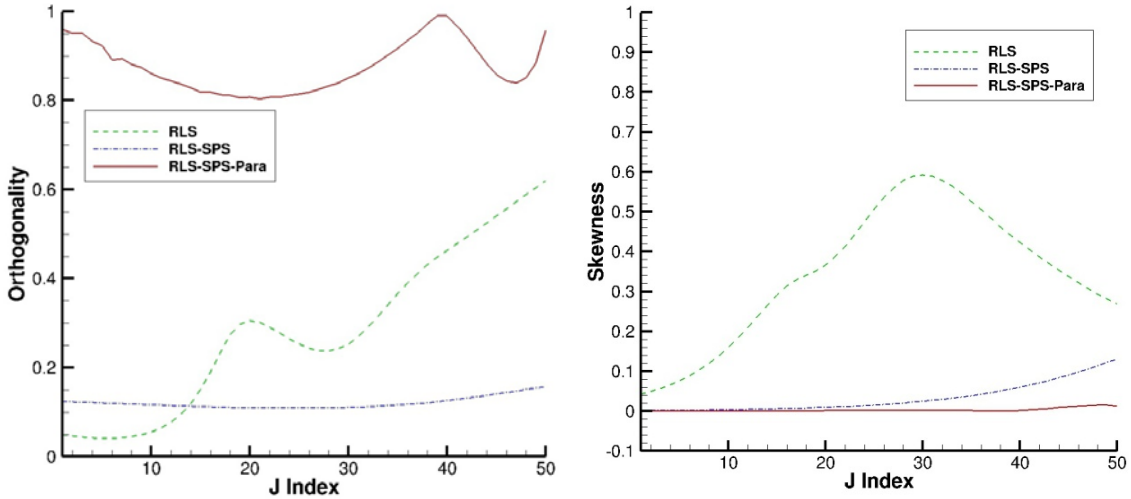

Figure 5.17: Grid quality comparison on i=148 (for (1≤ j ≤50): Orthogonality (left), Skewness (right).

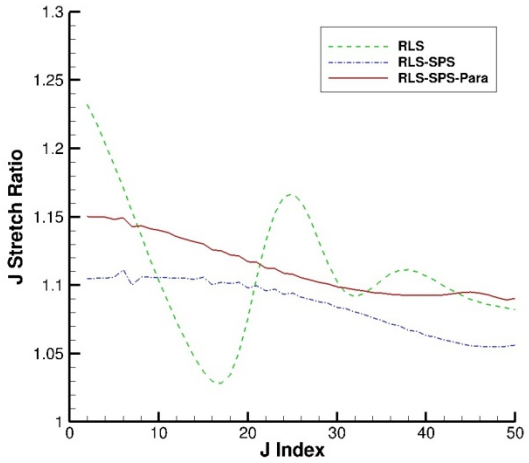

Figure 5.18: J Stretch Ratio comparison on i=148 (for (1≤ j ≤50).

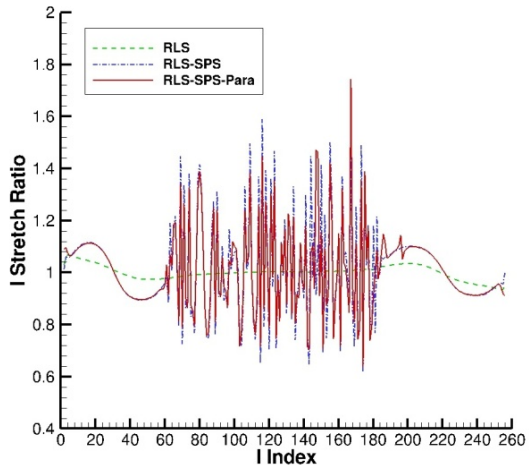

Figure 5.19: I Stretch Ratio comparison on j =50 (for 1≤ i ≤257).

A study has been done on the solution approaches used for solving the elliptic grid generation with the blended source terms (RLS-SPS-Para). Figure 5.20 shows all the Single-Grid solution convergence for ADI (in I, J, and IJ), Line SOR (LS) (in I, J, and IJ), Point SOR (PS), Point Jacobi (PJ), and Gauss Seidel (GS). Note that for ADI and Point Jacobi, the elliptic equation has been linearized using Newton Raphson approach, while for the rest, the successive substitution is applied. The factor ω is the SOR relaxation factor and dt is the implicit scheme time stepping, which is adjusted to the maximum possible value to reduce the computation time. The order of magnitude of dt is 10^{-2} to 10^{10}. Sweeping forward, backward or back and forth sometimes shows different stability behaviors, but a single forward sweep is found to be the most efficient. The Line SOR in I (LS-I) algorithm provides the best convergence time. ADI solution in I (ADI-I) also shows good convergence properties**.**

Figure 5.21 shows the effect of Multi-Grid application on the equation solved by LS-I and ADI-I. Only 2 levels of Multi-Grid V-cycle have been used with 20 relaxing steps on the coarse mesh, which shows reduction of the computation time by 48% for LS-I and by 44% for ADI-I. Note that the source terms are only computed on the fine mesh, with simple restriction to the coarse mesh.

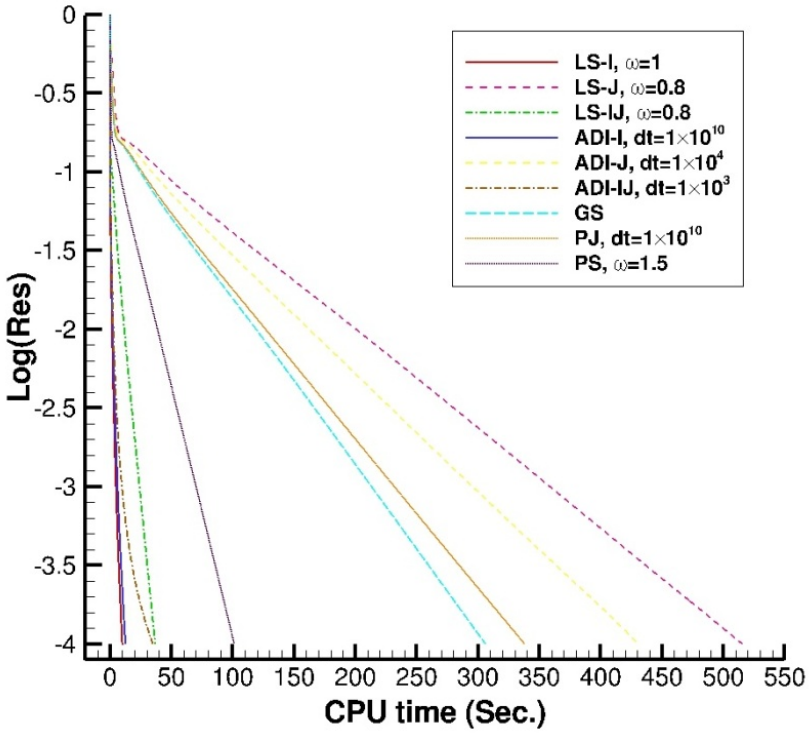

Figure 5.20: Solver comparison on the elliptic equation with blended source term (RLS-SPS-Para).

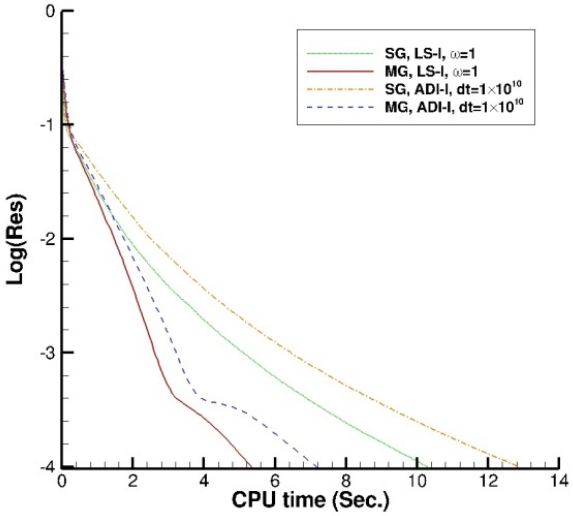

Figure 5.21: Multi-grid effects comparison on ADI-I and LS-I.

The effect of the linearization method for the quasi-linear elliptic equation of grid is now examined. Figure 5.22 shows the comparison of the linearization methods, Newton-Raphson (NR) and Successive Substitution (SS), for ADI solution approach. The factor dt is adjusted to the maximum value for stability and computation time reduction. It has been observed that the ADI in I with Newton-Raphson linearization (NR) has significantly better convergence compared to Successive Substitution linearization (SS) technique. Notably, the ADI-I (NR) scheme allows dt to be at 10^{10}.

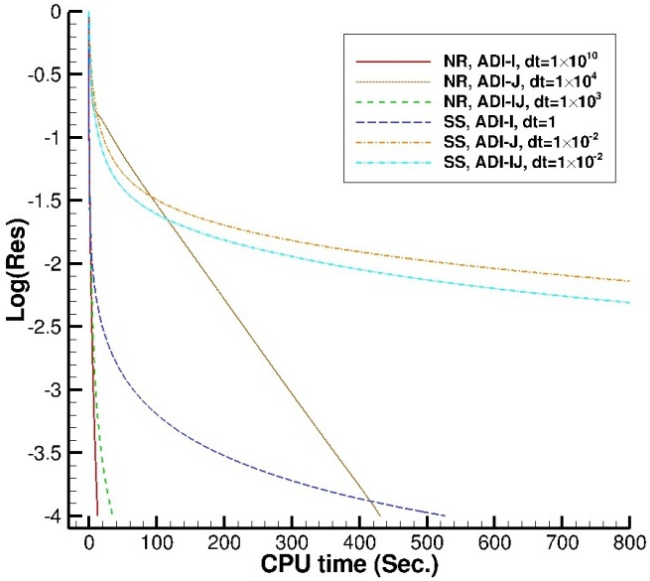

Figure 5.22: Linearization comparison, NR and SS, on ADI-I.

5.4 Standard validation cases results

In this section, different simulation cases are presented with the objective to validate the flow solution simulation on the generated structured grids. The selected test cases are a clean NLF0414 airfoil and an experimental glaze ice 623 on NLF0414 airfoil. All the results in this paper were produced using the NSCODE solver to solve the flow on the structured grids generated with the developed elliptic ice grid generation package NSGRID.

5.4.1 Navier-Stokes solver

The Navier-Stokes flow solver NSCODE, a finite volume two-dimensional multi-block Euler/Navier-Stokes flow solver with Full Approximate Storage Multi-Grid developed at Ecole Polytechnique [94], was also incorporated within the CANICE icing framework [20, 31, 63].

Implemented turbulence models include the widely used Spalart-Allmaras and k-ω-SST, γ-R_θ equations. NSCODE also includes a wall treatment roughness Boeing model which has been implemented and validated with the Spalart-Allmaras turbulence model.

5.4.2 Standard validation cases

Two case studies are now analyzed for NSGRID validation, the clean NLF0414 airfoil and the experimental glaze ice 623 on NLF0414 airfoil, both at Mach 0.29 and Reynolds 6.4×10^6. The clean and experimental ice geometry data are given in reference [48]. The case has been analyzed in the literature using SmaggIce grid generation (with multi-block structured grids) and WIND CFD solver to compute the aerodynamic performances [6] and in [7, 115, 116] using ICEG2D grid generation (with structured grid) and NPARC solver to compute the aerodynamic performances. Note that the solution of WIND CFD solver is with the k-ω-SST turbulence model and the solution of NPARC solver is with Spalart-Allmaras turbulence model.

For the grid generation process using NSGRID, the surface mesh generation solver uses Point SOR (convergence to 1×10^{-4}) with source term coefficient $A=2\times10^{-4}$ and the curvature section criterion of 0.1 (for the glaze ice shape). The field mesh generation solver uses the blended approach (RLS-SPS-Para) with weight factors of 1, 0.5 and 0.2 (corresponding to RLS, SPS and Para source terms, respectively) and Line SOR in I with 2 levels V-cycle Multi-Grid (convergence to 1×10^{-4}).

5.4.2.1 Clean NLF0414 airfoil

Using NSGRID, the generated elliptic grid (257 by 129) using 1D PDE elliptic curvature based surface point's distribution and blended approach (RLS-SPS-Para) is shown in Figure 5.23.

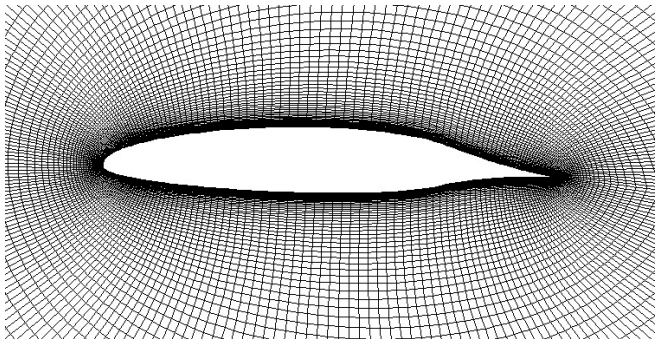

Figure 5.23: Elliptic grid generated with blended approach RLS-SPS-Para.

The grid convergence solution is shown in Figure 5.24, using Line SOR in I with 2 levels of Multi-Grid. The computed aerodynamic performances versus angle of attack, Lift and Drag coefficients are shown in Figures 5.25 and 5.26, respectively. Results show good stall prediction compared to the numerical and experimental data. It is observed that the k-ω turbulence model results in better flow prediction compared to Spalart-Allmaras model, especially in C_{Lmax} prediction. All cases are run by NSCODE with 5 levels of Multi-Grid (ij coarsening) and JST artificial dissipations of 1/4 (second order dissipation), and 1/128 (fourth order dissipation) [117]. The roughness model is not activated for the case and the surface is assumed to be smooth. In the case of unsteady oscillations occurring at high angle of attack, aerodynamic results are averaged in time. The flow convergence for an angle of attack of 0°, and different turbulence models, Spalart-Allmaras (S-A) and k-ω, is shown in Figure 5.27.

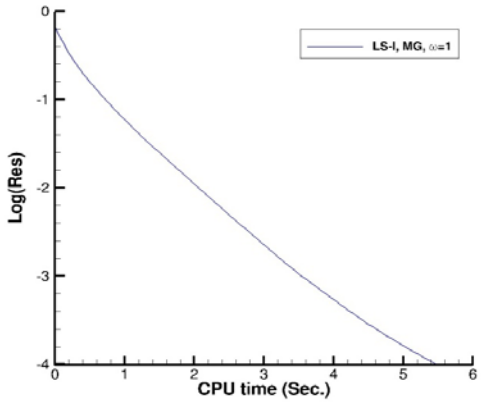

Figure 5.24: Grid solution convergence of the blended source term (RLS-SPS-Para), using LS-I, with 2 levels of Multi-Grid.

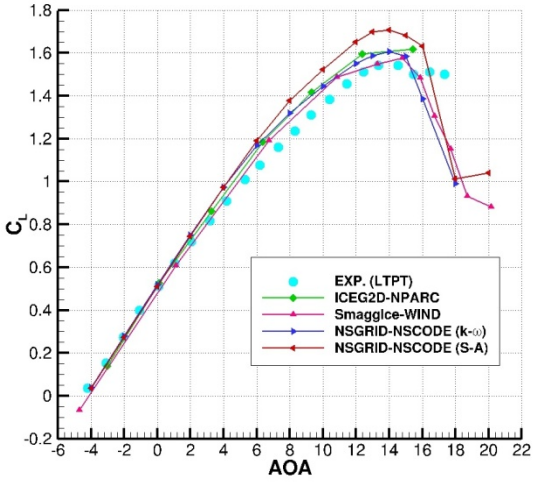

Figure 5.25: Lift coefficient versus angle of attack.

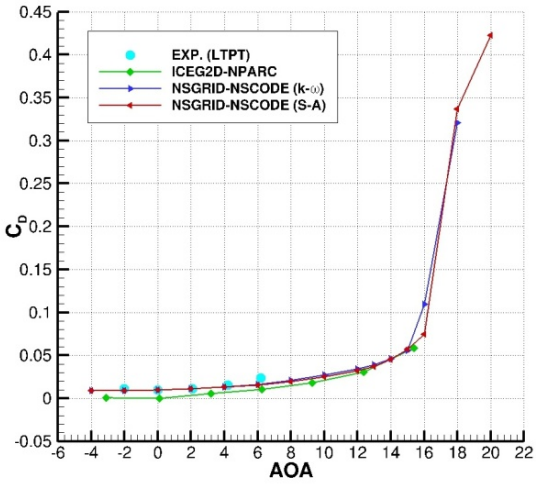

Figure 5.26: Drag coefficient versus angle of attack.

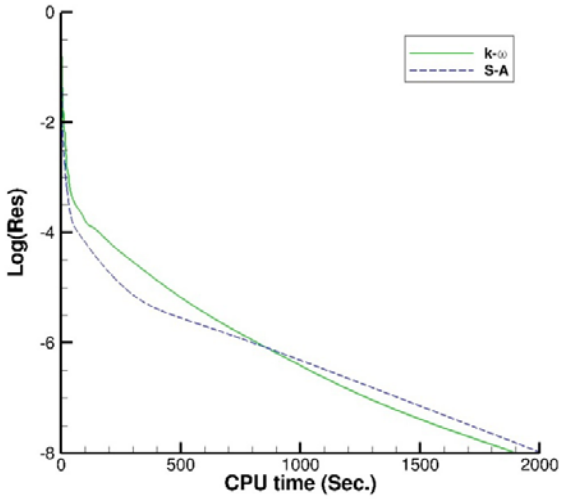

Figure 5.27: NSCODE flow convergence comparison for AOA of 0°.

5.4.2.2 Experimental glaze ice case 623-2D on NLF0414 airfoil

The experimental ice geometry data is given in reference [48]. Using NSGRID, the generated elliptic grid (513 by 129) using 1D PDE elliptic curvature based surface point distribution and blended approach (RLS-SPS-Para) is shown in Figure 5.28. The computed ice surface curvature is shown in Figure 5.29 with selected points with curvature higher than 0.1 (user input) in green (triangle). The elliptic curvature based grid method is applied on the surface with coefficient $A=2\times10^{-4}$ for the surface grid points source term. The computed curvature based points are shown in Figure 5.29, based on the point spacing along the ice surface geometry versus point arclength and shows the concentration of the grid around selected points with positive curvature.

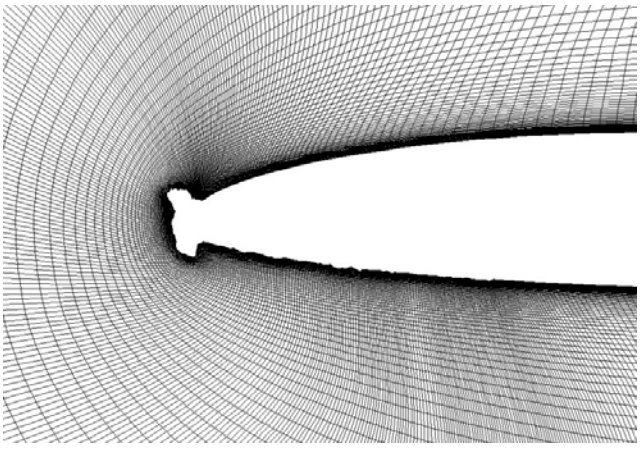

Figure 5.28: Elliptic grid generated with blended approach RLS-SPS-Para.

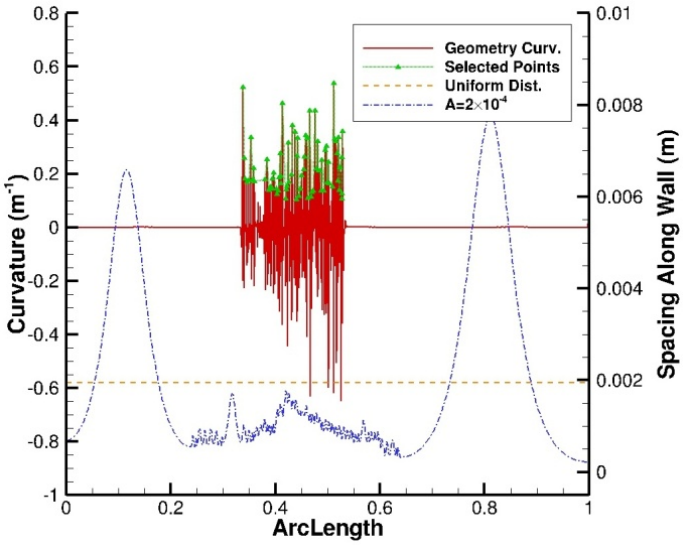

Figure 5.29: Computed curvature (left axis: geometry curvature in red, selected points in green); Curvature based distributed points spacing along the wall (right axis: for coefficient $A=2\times10^{-4}$).

The NSGRID convergence solution is shown in Figure 5.30, using Line SOR in I with 2 levels V-cycle Multi-Grid. The computed aerodynamic performances versus angle of attack, Lift and Drag coefficients are shown in Figures 5.31 and 5.32, respectively. Using S-A turbulence model results in higher lift coefficient compared to k-ω model as shown in Figure 5.31. All the cases are run by NSCODE with 5 levels of Multi-Grid (j coarsening) and JST artificial dissipations of 1/2 (second order dissipation), and 1/80 (fourth order dissipation). Because of the complexity of the ice shape geometry ij coarsening has shown stability issues, but j coarsening has solved the problem. In the case of oscillations at high angle of attack, aerodynamic results are averaged in time.

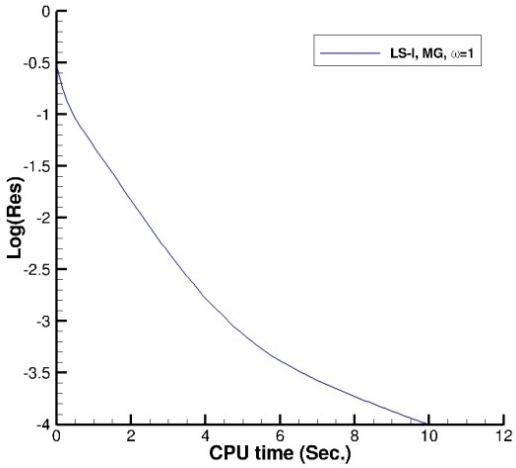

Figure 5.30: Grid solution convergence of the blended source term (RLS-SPS-Para), using LS-I, with 2 levels of Multi-Grid.

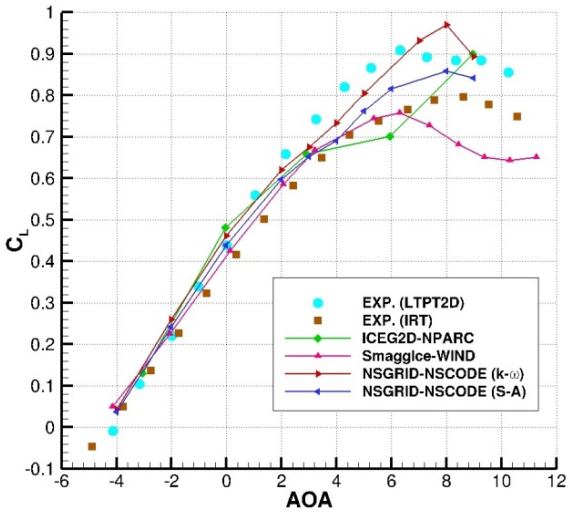

Figure 5.31: Lift coefficient versus angle of attack.

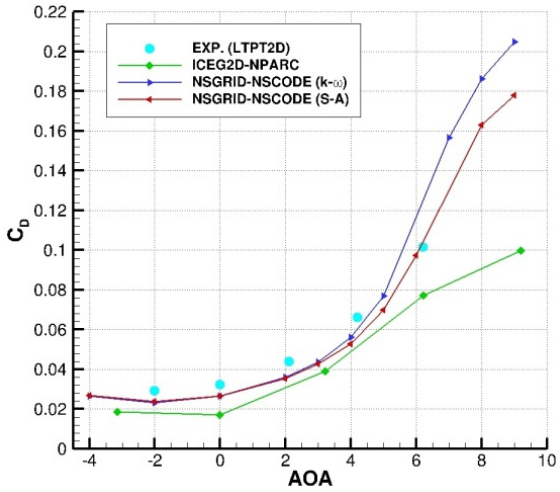

Figure 5.32: Drag coefficient versus angle of attack.

The pressure coefficient comparison for the angle of attack of 0° is shown in Figure 5.33. The flow convergence for different turbulence models, Spalart-Allmaras (S-A) and k-ω, is shown in Figure 5.34. Note that the flow solution with the k-ω model indicates the presence of unsteady flow patterns. The roughness model is not activated for the case and the surface is assumed to be smooth. The predicted lift-curve by NSGRID-NSCODE (elliptic structured grid) shows good results compared to ICEG2D-NPARC [115] (parabolic structured grid) results and SmaggIce-WIND [6] (multi-block structured grid) results. The predicted lift coefficient is in line with two experimental data sets (LTPT2D and IRT) [6, 7, 115]. Also, the predicted drag coefficient by NSGRID-NSCODE presents good comparison with the experimental data and is better than the ICEG2D-NPARC (parabolic structured grid) solution [115].

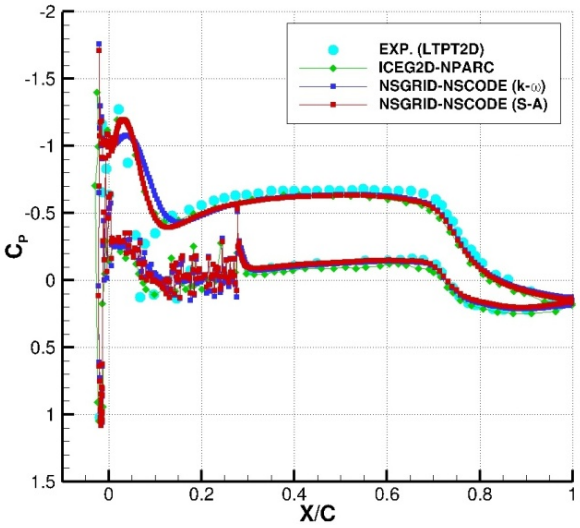

Figure 5.33: Cp comparison for AOA of 0°.

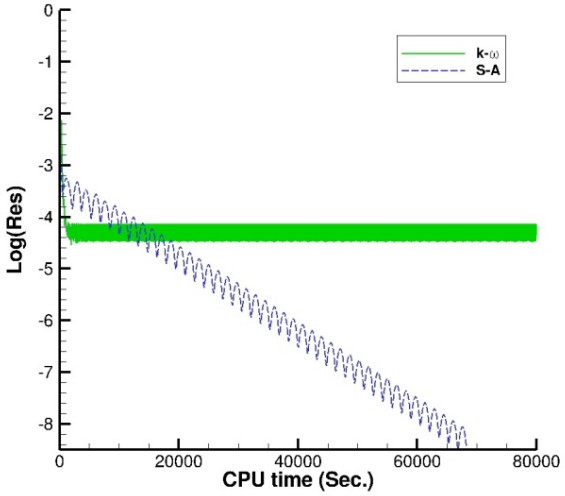

Figure 5.34: NSCODE flow convergence comparison for AOA of 0°.

5.5 Conclusion

In this paper, a novel grid generation method has been proposed to handle both surface and field mesh generation around complex glaze ice with convex/concave regions. First, a novel elliptic curvature based surface point distribution scheme treating convex/concave regions is presented. Second, a blended approach has been developed in the context of elliptic grid generation via appropriate choice of source terms. Results show the improvements of the proposed blended approach (RLS-SPS-Para) compared to other classical methods (Sorenson, Spekreijse and parabolic) in resolving complex glaze ice shapes by carefully describing the role of the source terms properties (spacing, curvature, orthogonality) and examining their effects on grid metrics (wall spacing, orthogonality, stretching ration, skewness). Solution of the resulting partial differential equation is analyzed in the context of a full-multigrid operator, showing the superior convergence rate of the Line SOR-I and ADI-I algorithms. Finally, a demonstration of the applicability of the mesh generation procedure to standard icing cases is made by comparing Navier-Stokes solutions using École Polytechnique NSCODE with results of NASA's SmaggIce-WIND and ICEG2D-NPARC packages.

CHAPTER 6

UPGRADE OF MULTI-STEPS RANS BASED ICE ACCRETION FRAMEWORK OF CODE CANICE2D-NS

Here we present the newly upgraded framework of CANICE2D-NS using first: a new RANS flow solver NSCODE2D (replacing NSMB3D solver) and second, a new grid generation solver NSGRID2D (replacing ICEM-CFD meshing tool). The code is validated by a variety of standard icing cases to study the robustness of the framework that is considered here by the number of successful computations.

6.1 Upgrade of RANS solver

The NSCODE2D RANS solver contains a very efficient Full Multigrid algorithm with an explicit multistage Runge-Kutta solver which differs from the single grid LU-SGS implicit solver in NSMB. Here, the effects of the Full Multigrid operator and an implementation of a J-coarsening strategy are examined.

6.1.1 NSCODE2D RANS solver

NSCODE2D is a finite volume two dimensional multi-block Euler/Navier-Stokes based solver with Full Approximate Storage Multi-Grid developed by the laboratory of prof. Laurendeau [31, 94]. Available turbulence models include the Spalart-Allmaras, k-ω-SST and γ-R_θ equations. The code incorporates the wall treatment roughness model of Boeing which is implemented and validated within the Spalart-Allmaras turbulence model. NSCODE2D is capable of steady and unsteady flow analysis and incorporates a Chimera method. The code is verified and validated on a variety of steady and unsteady case studies [45, 63].

A J-coarsening multigrid approach, as examined by Giles et al. [118], was implemented in the solver by simply performing the coarsening operations in the wall normal direction (J), while leaving the curvilinear direction (I) unchanged. The effect is to retain the icing surface geometry on the coarse meshes while removing stiffness in the boundary-layer direction.

6.1.2 Icing validation results

Here a number of case studies are considered to validate first, NSCODE RANS solver and second, the new framework of CANICE2D-NS. The standard cases for NSCODE include: a flat plate cases for roughness model analysis, a transonic airfoil RAE2822 for aerodynamic performance analysis and a glaze ice case for FMG-J coarsening analysis. The standard cases for CANICE-NS framework include a rime ice run 405 and a glaze ice run 408.

6.1.2.1 Validation of NSCODE solver

The Boeing roughness model is a new addition to NSCODE, implemented by a Post-doctoral student (Dr. Ali Mosahebi) but validated in the context of this thesis.

Rough wall turbulent skin-friction over flat plate

The selected cases study for the Boeing rough wall model validation is the case of Hellsten and Laine [34], flat plate with the specifications: Length (or Chord) of 1 m, Reynolds 5 Million, Mach 0.2, and non-dimensionalised roughness height K_s (k_s/L or k_s/C) of (0.0001, 0.00025, 0.0005, 0.001 and 0.0015). The generated mesh is shown in Chapter 4, as it was used for NSMB Rough wall ONERA model validation. The comparison of NSCODE skin friction solution, using Spalart-Allmaras turbulence model with Boeing extension, with the with the Mills and Hang [34] semi-empirical relation (cited in Chapter 4) is shown in Figure 6.1, which shows a good agreement.

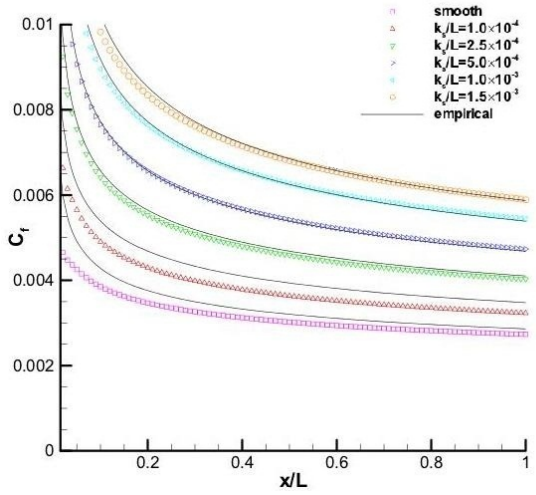

Figure 6.1: Turbulent flat plate skin friction comparison: NSCODE (SA-Boeing) with semi-empirical relation.

2D transonic RAE2822 airfoil

The second validation cases is a 2D transonic airfoil RAE2822 [35] with the following specifications: AOA of 2.31°, Mach 0.729, Reynolds 6.5 Million [36]. The generated grid is shown in Chapter 4, as it is it was used for NSMB solver validation. The solution convergence comparison between NSMB and NSCODE is shown in Figure 6.2. Note that NSCODE is using 5 levels FMG operator compared to NSMB Single-Grid solution that results shows over 14 orders reduction. The pressure coefficient comparison for both NSMB and NSCODE codes and the literature data (cited in Chapter 4 as part of NSMB validation process) are shown in Figure 6.3, proving the accuracy of the NSCODE solver.

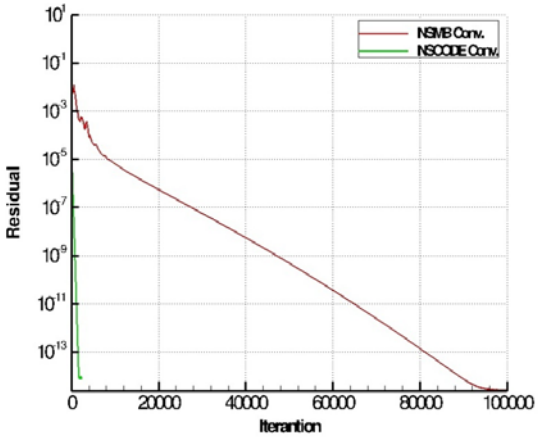

Figure 6.2: Residual Convergence, NSMB and NSCODE on the RAE2822 airfoil.

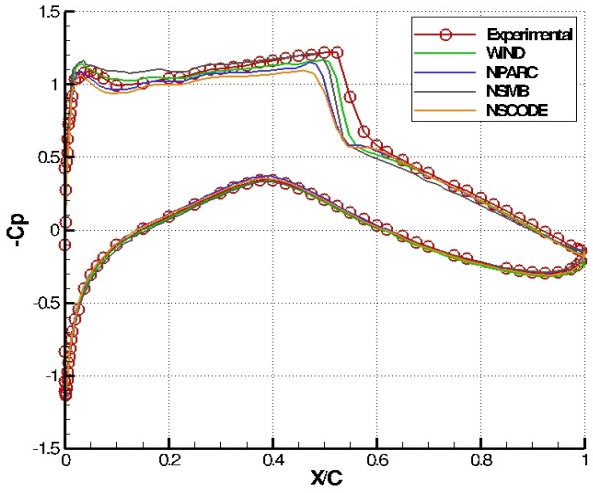

Figure 6.3: Pressure coefficients comparison for the RAE2822 airfoil.

FMG solution for glaze ice

Application of FMG-J coarsening has shown increase in RANS flow solution robustness for complex glaze ice form problems. The generated mesh is shown in Figure 6.4. It is seen that the I-coarsening of FMG operator results in interpolation errors over the complex ice forms which finally cause computation divergence. The solution convergence rates for single-grid, FMG and FMG-J are shown in Figure 6.5, proving the FMG-J coarsening efficiency. Therefore, in this work, FMG-J coarsening operator is used for increase in robustness of the multi-time steps RANS based ice accretion framework of CANICE2D-NS.

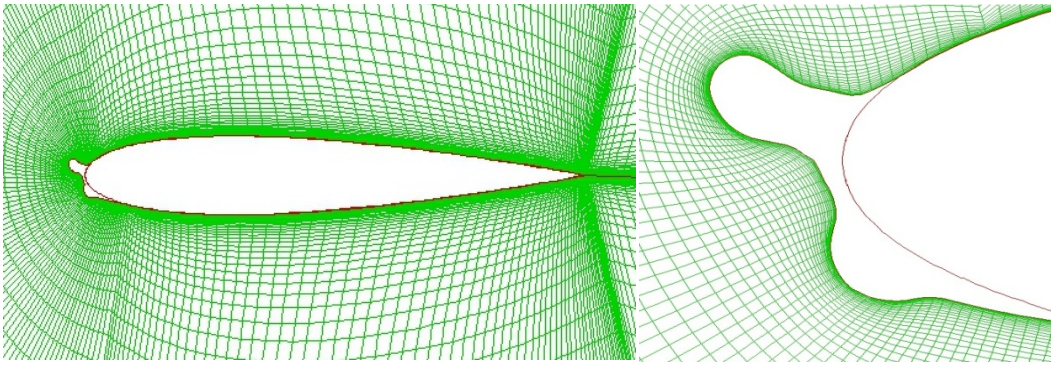

Figure 6.4: NACA0012 run 408 glaze ice C-mesh using automated ICEM grid generation.

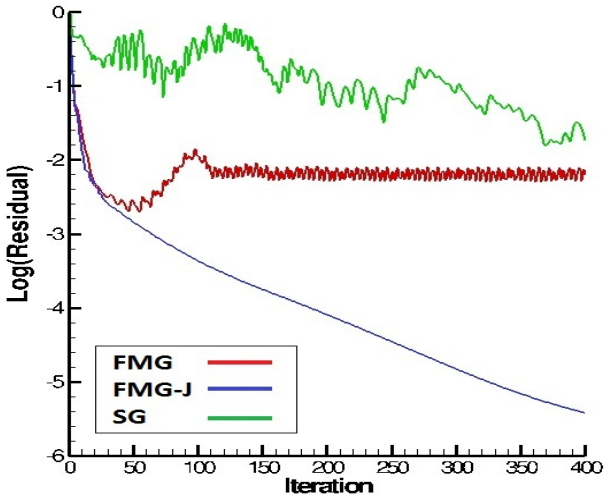

Figure 6.5: Convergence rates comparison of SG, FMG, and FMG-J.

6.1.2.2 Validation of CANICE2D-NS framework

Two standard icing cases, a rime ice run 405 and a glaze ice run 408 are considered. The case studies data are cited in Chapter 4, as they were used for the preliminary icing frame work validation of (CANICE2D-NS with NSMB solver). The collection efficiency comparison is

shown in Figure 6.6. The ice shapes growth is effected by different roughness values and the ice shapes comparisons with literature data are shown in Figures 6.7 and 6.8, respectively. Note that the K_s value of 0.005 results in the best match to the experimental ice shape. The NSCODE RANS solver uses the Spalart-Allmaras turbulence model with Boeing rough wall model correction. The FMG-J coarsening operator is used in NSCODE for solution convergence improvement and acceleration. The convergence comparison of NSCODE FMG-J with NSMB Single-Grid is shown in Figure 6.9, which demonstrates over 10 time reduction in number of iterations for similar residuals reductions.

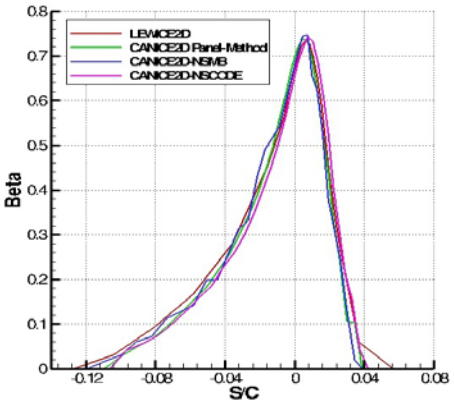

Figure 6.6: Collection efficiency comparison.

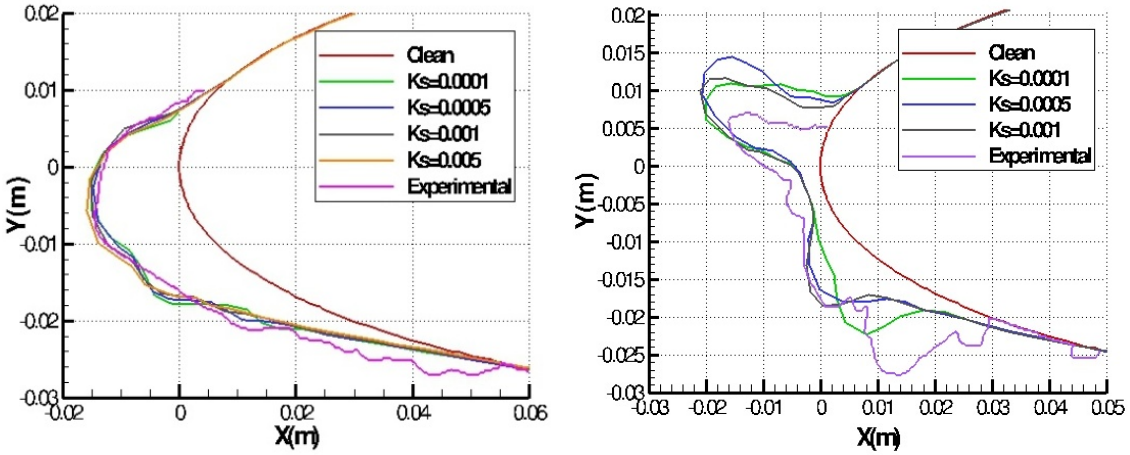

Figure 6.7: Ice shape comparison for different roughness, CANICE2D-NSCODE, run 405 (left), run 408 (right).

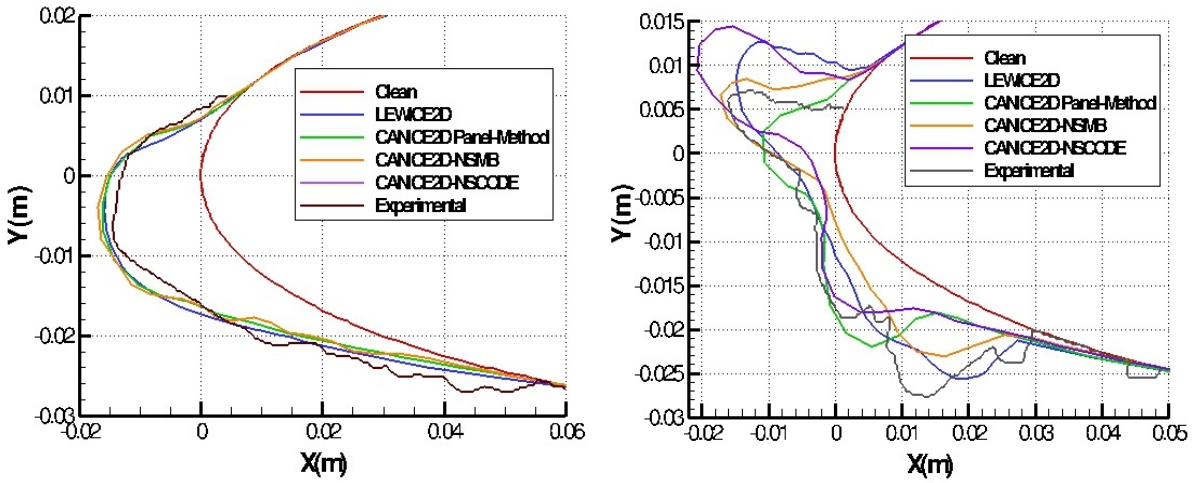

Figure 6.8: Ice shape comparison with literature data, CANICE2D-NSCODE, run 405 (left), run 408 (right).

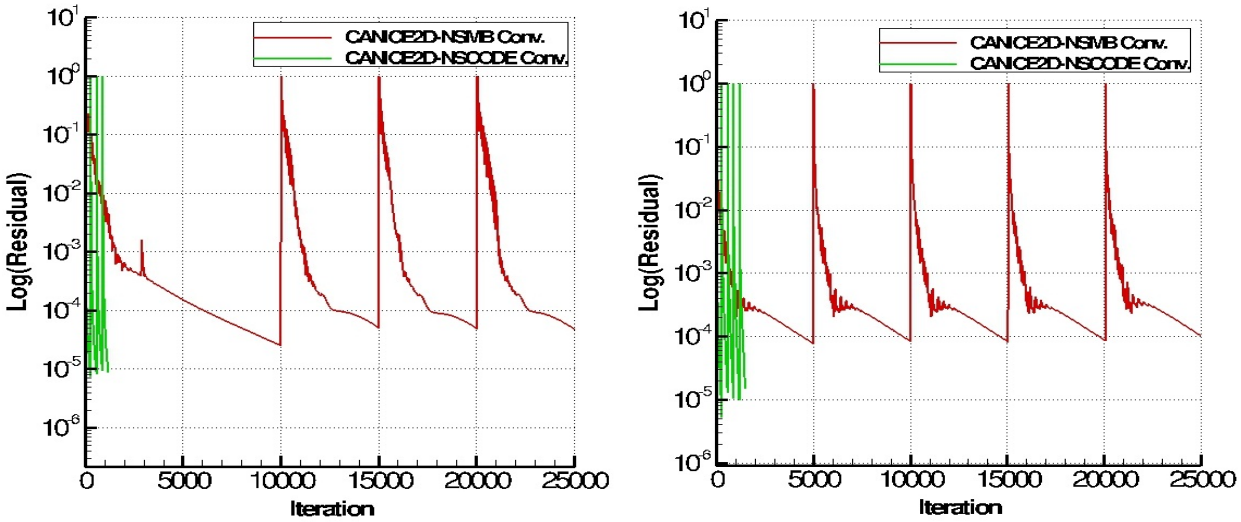

Figure 6.9: Multi-steps convergence of CANICE2D-NSBM compared to CANICE2D-NSCODE, run 405 (left), run 408 (right).

6.2 Upgrade of grid solver

The newly developed grid generation code NSGRID2D, described in Chapter 5, is coupled with the framework of CANICE2D-NS (replacing ICEM-CFD mesh solver) with NSCODE RANS solver. The framework is validated by a variety of standard NATO test cases.

6.2.1 Icing validation results

Since the grid generation solver is validated in Chapter 5 and NSCODE in validated in [94] we present here the validation of the icing framework of CANICE2D-NS coupled with NSGRID2D grid solver and NSCODE2D flow solver. Focus is the analysis of the robustness of the framework and its limitations. The icing framework robustness is considered by the number of successful case studies computation out of the total number of cases.

The NATO cases studied here include: C09, C10, C13, C14, C15, C16, C17, and C18 [9], which are cited in literatures for validation of LEWICE-Panel, CANICE-Panel, and CANICE-BA-Panel (v3.5) codes [20, 119]. The cases data are shown in Table 6.1. Three choices for number of time steps are chosen: Single-time (TS=1), 5-time steps (TS=5), and 10-time steps (TS=10). Three choices for roughness value K_s are chosen (0.0001, 0.0005 and 0.001). Therefore the total number of case runs is equal to 72.

The NSGRID2D solver properties are: O-mesh topology with 257 and 129 points in i and j respectively, blended elliptic solver with weight factors of (f_{RLS}=1, f_{SPS}=0.5, and f_{ortho}=0.001), ADI-I solver, 1st cell height of (y^+<1), and stretching ratio less than 1.15. The NSCODE2D flow solver properties are: Runge-Kutta scheme with 4 levels FMG-J coarsening, JST dissipation coefficients of 1/2 and 1/32, and CFL=7.5. The CANICE2D-NS solver uses a Lagragian droplet computation and a traditional Messinger model ice thermodynamics computation. Note the grid, flow and ice solvers properties are decided in a sense to relax the icing problem and increase the framework robustness.

Figure 6.10 shows the typical grid (NSGRID2D) and flow (NSCODE2D) solution convergence rate of 5 time-steps ice accretion calculation for the cases (here shown C09). Figure 6.11 shows the typical effects of roughness values on 5 time-steps ice accretion formation for the cases (here shown C09).

The essential results comparisons for all the test cases are shown in Figures 6.12-6.27 to emphasis the test cases successful computations. Note that based on the literature data, the number of time steps applied in CANICE-BA (v3.5) code in the comparisons are as follows: 2-steps for the cases C9, C10 and C14 and 3-steps for the cases C15 to C18.

We note that glaze ice test cases can have ice horns so large that the flow solver steady state convergence is not reached, as shown in Figure 6.28. The convergence pattern indicates the present of unsteady flow effects that cannot be captured by the current steady RANS approach.

Table 6.1: NATO cases test conditions.

Cases	**C09**	**C10**	**C13**	**C14**	**C15**	**C16**	**C17**	**C18**
Airfoil	NLF0414	NLF0414	NACA0012	NACA0012	SA13112	SA13112	NACA0012	NACA0012
Chord (m)	0.9	0.9	0.53	0.27	0.6	0.6	0.91	0.91
AOA	0	3.5	0	0	10	0	3	3
Speed (m/s)	92.54	67.04	67	57	81.3	163	67	67
Temp. (K)	257.59	262.73	264.39	267.59	263.19	263.19	265.99	265.99
Mach	0.2876	0.2063	0.2055	0.1738	0.2499	0.5012	0.2049	0.2049
Rey.	6.95E6	4.86E6	2.82E6	1.19E6	3.91E6	7.85E6	4.80E6	4.80E6
LWC (g/m^3)	0.33	0.44	0.65	1.04	0.6	0.6	1	1
MVD (μm)	20	20	40	27.73	20	20	24.8	38.8
Time (min.)	20.4	22	11.2	4.12	15	7.5	20	20

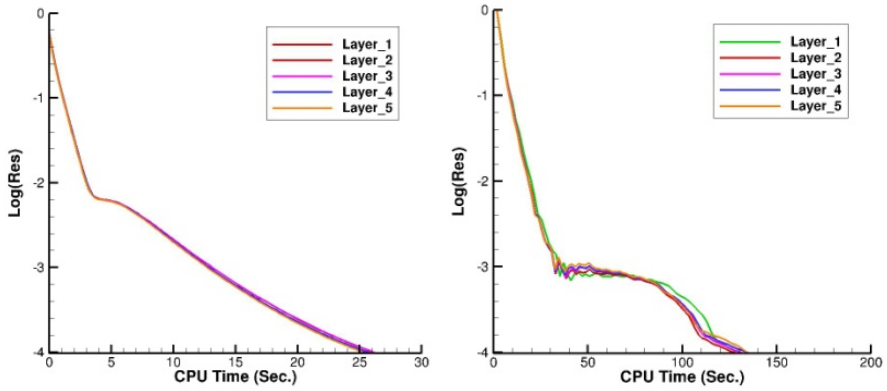

Figure 6.10: Convergence rate, NSGRID2D (left), NSCODE2D flow (right), (case C09, TS=5).

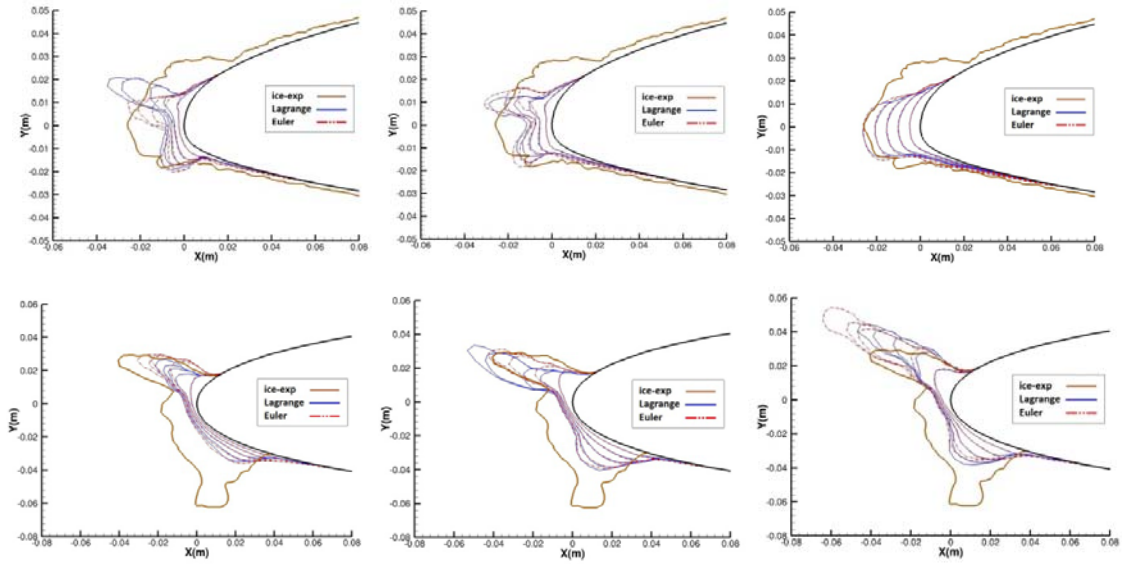

Figure 6.11: Ice shape formation effected by roughness for case with 5 time steps: case C09 rime ice (top), case C17 glaze ice (down); Ks =0.0001 (left), Ks =0.0005 (middle), Ks =0.001 (right).

The collection efficiency comparison and the ice shape comparison for case C09 are shown in Figure 6.12. The generated grid and flow for a test run are shown in Figure 6.13.

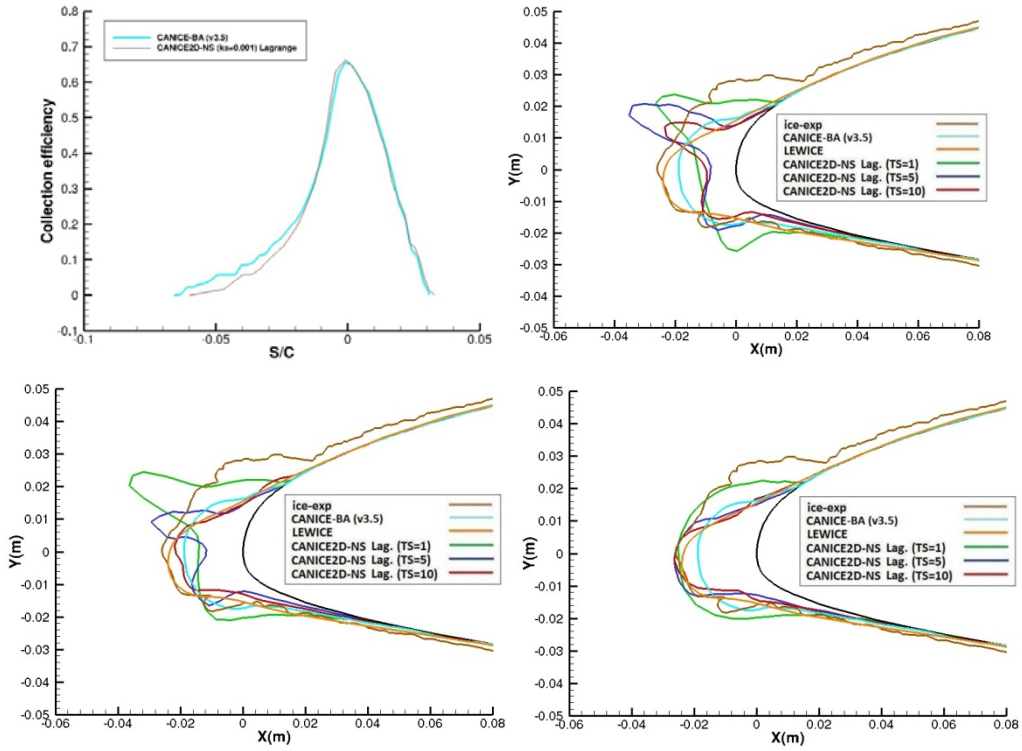

Figure 6.12: Collection efficiency comparison: Lagrangian method, top-left; Ice shape comparison: (Ks=0.0001), top-right; (Ks=0.0005), down-left; (Ks=0.001), down-right.

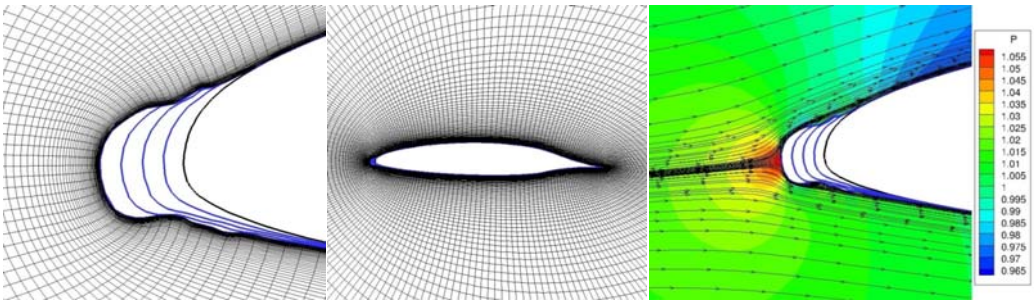

Figure 6.13: Generated grid and computed flow, (TS=5, Ks=0.001, case C09).

The collection efficiency comparison and the ice shape comparison for case C17 are shown in Figure 6.14. The generated grid and flow for a test run are shown in Figure 6.15.

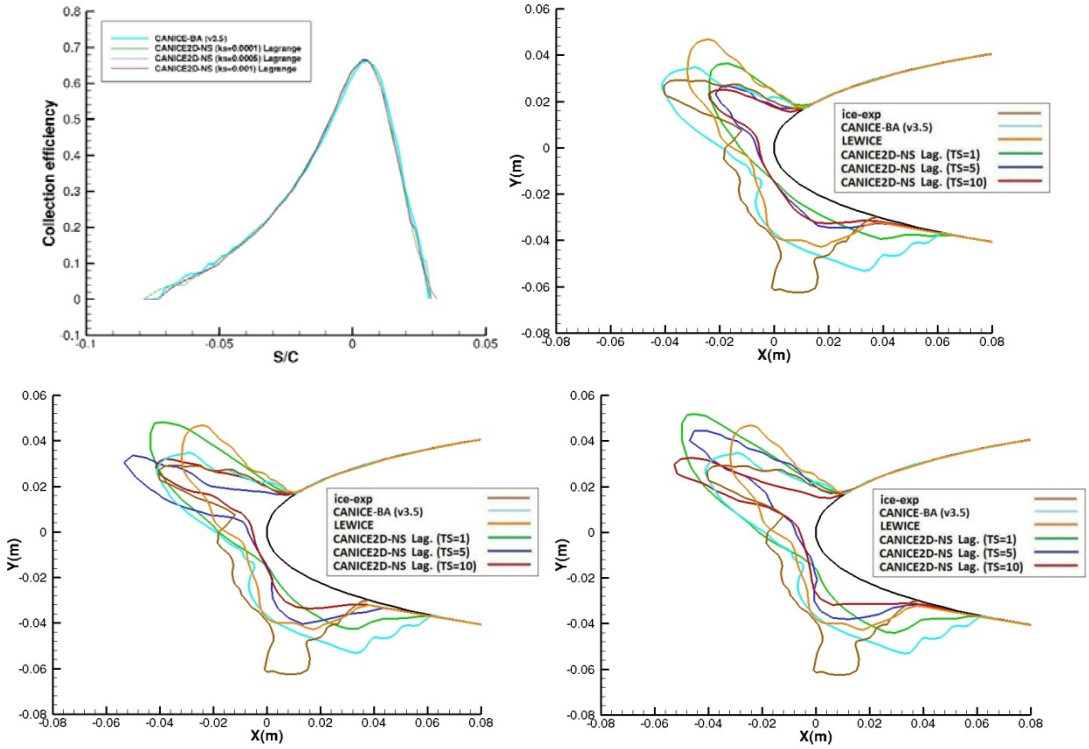

Figure 6.14: Collection efficiency comparison: Lagrangian method, top-left; Ice shape comparison: (Ks=0.0001), top-right; (Ks=0.0005), down-left; (Ks=0.001), down-right.

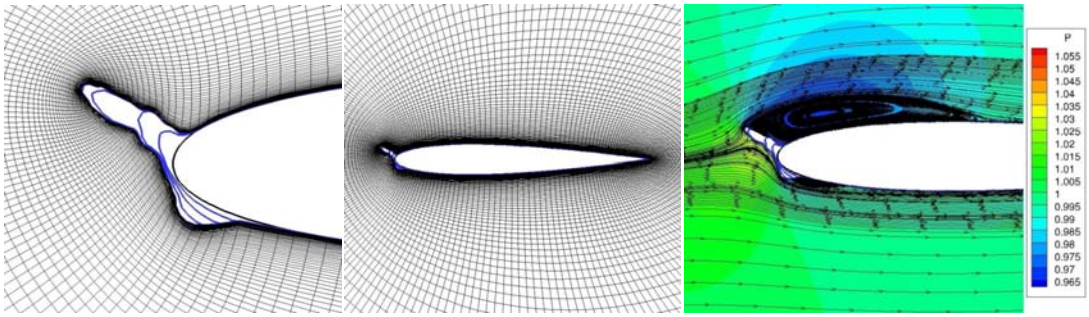

Figure 6.15: Generated grid and computed flow, (TS=5, Ks=0.001, case C17).

The collection efficiency comparison and the ice shape comparison for case C10 are shown in Figure 6.16. The generated grid and flow for a test run are shown in Figure 6.17.

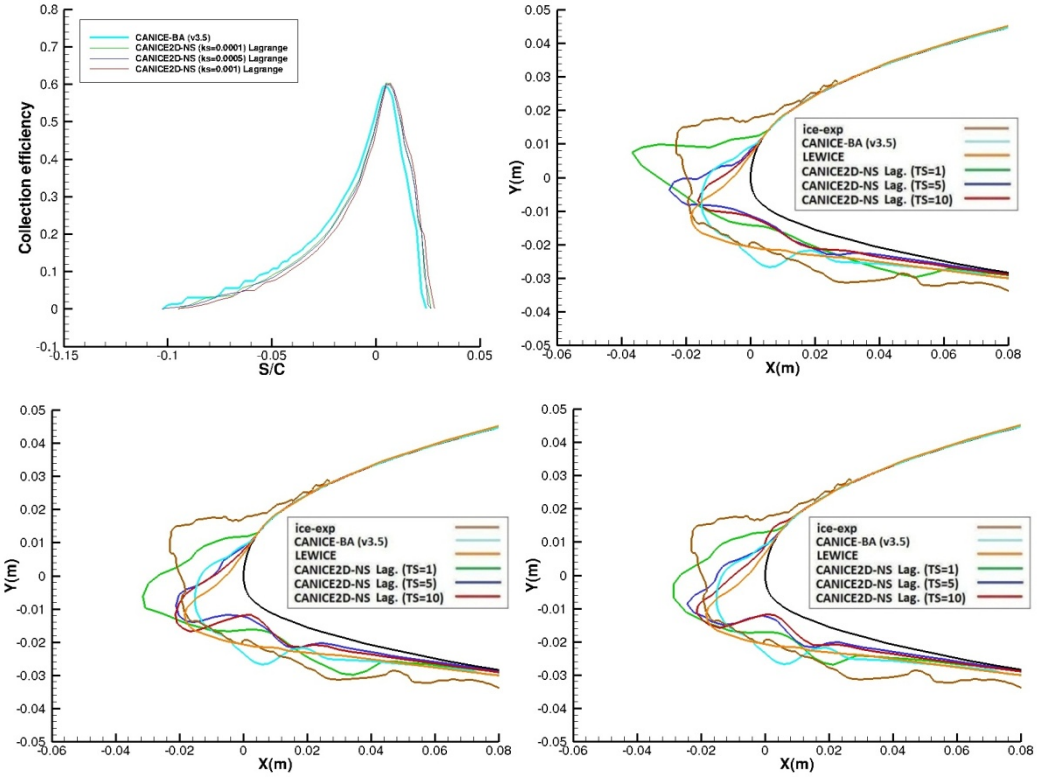

Figure 6.16: Collection efficiency comparison: Lagrangian method, top-left; Ice shape comparison: (Ks=0.0001), top-right; (Ks=0.0005), down-left; (Ks=0.001), down-right.

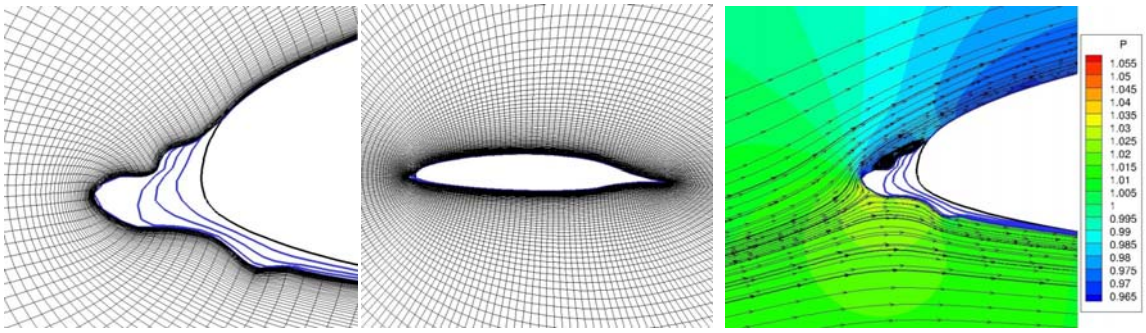

Figure 6.17: Generated grid and computed flow, (TS=5, Ks=0.001, case C10).

The collection efficiency comparison and the ice shape comparison for case C13 are shown in Figure 6.18. The generated grid and flow for a test run are shown in Figure 6.19.

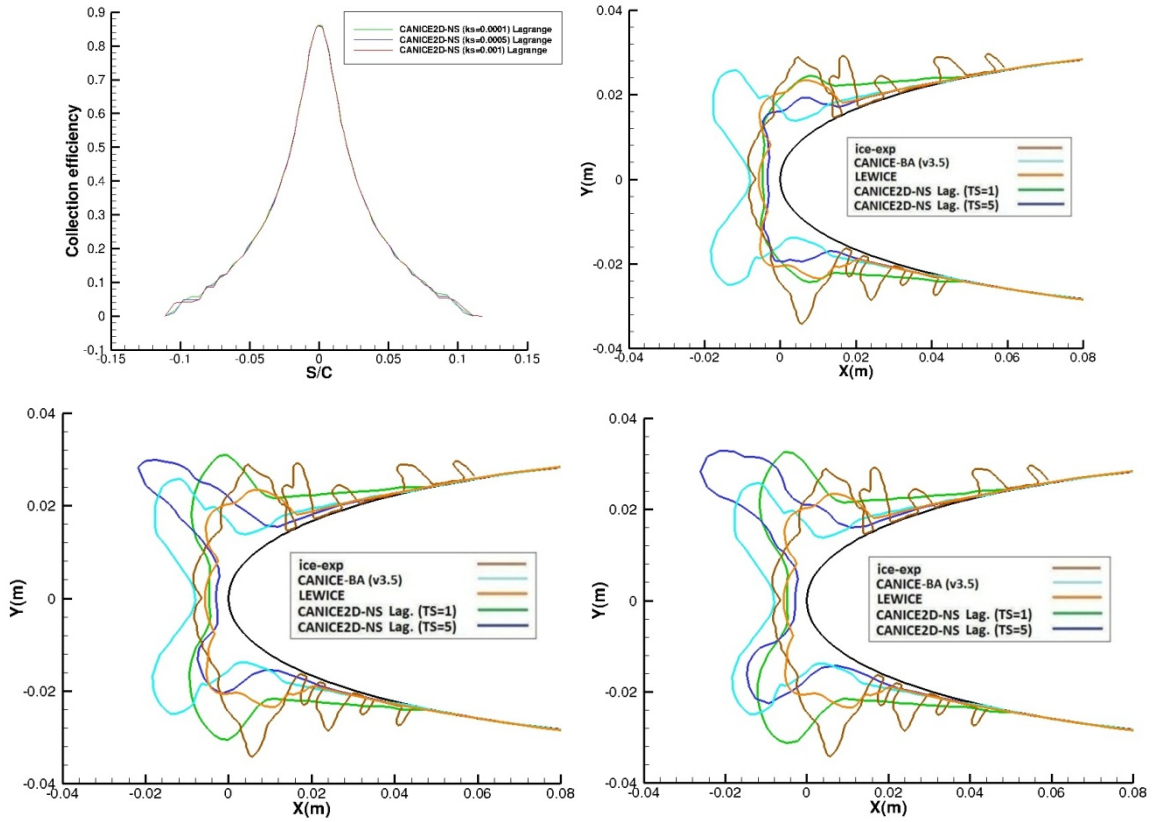

Figure 6.18: Collection efficiency comparison: Lagrangian method, top-left; Ice shape comparison: (Ks=0.0001), top-right; (Ks=0.0005), down-left; (Ks=0.001), down-right.

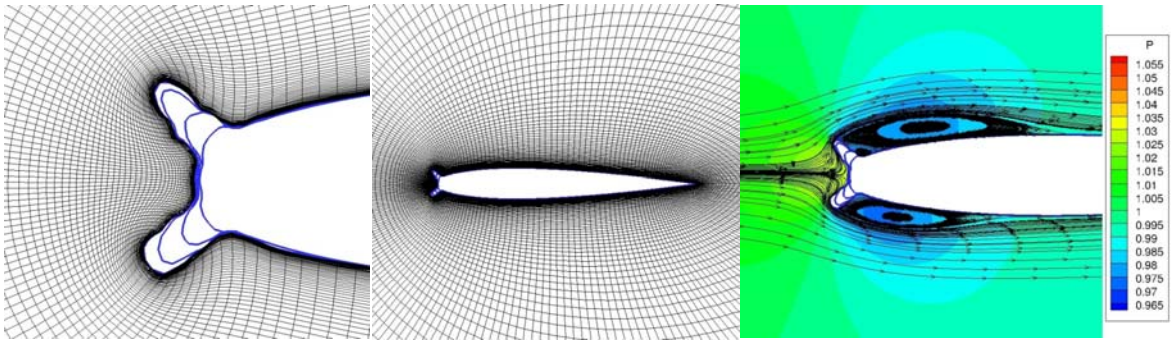

Figure 6.19: Generated grid and computed flow, (TS=5, Ks=0.001, case C13).

The collection efficiency comparison and the ice shape comparison for case C16 are shown in Figure 6.20. The generated grid and flow for a test run are shown in Figure 6.21.

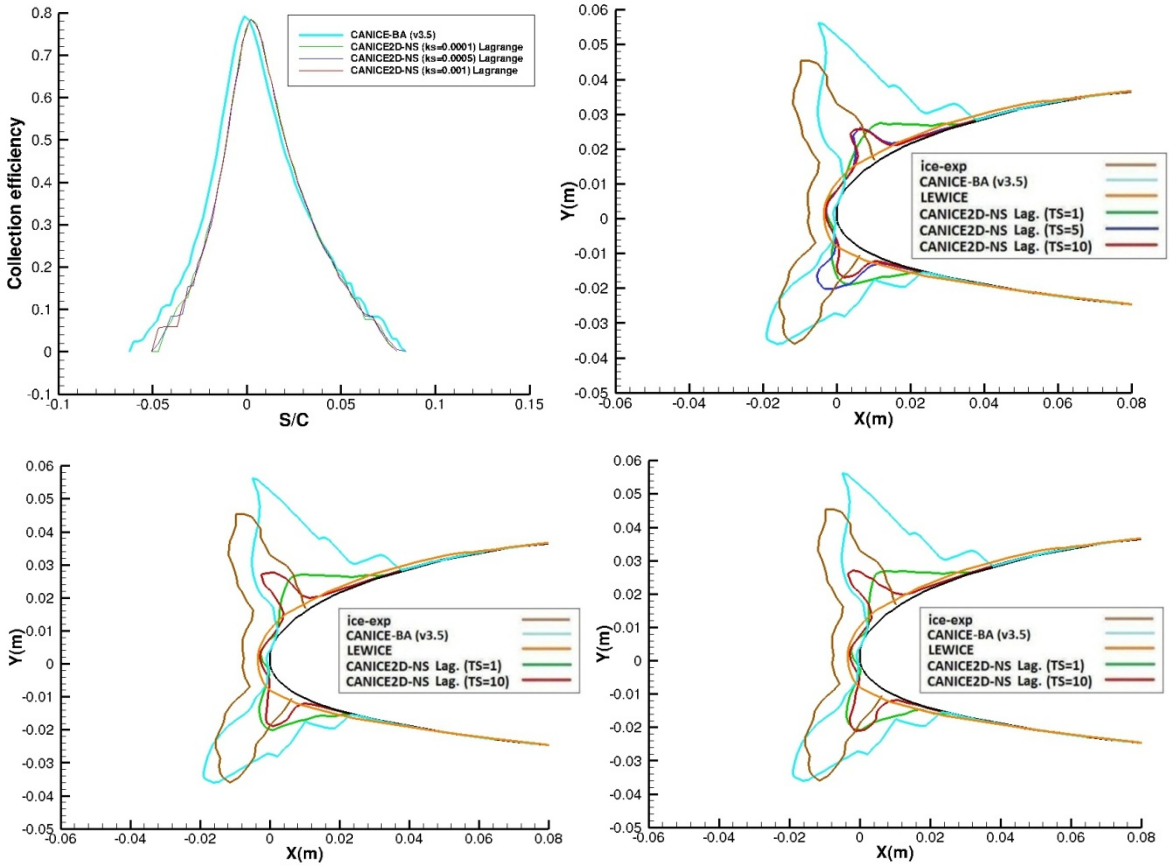

Figure 6.20: Collection efficiency comparison: Lagrangian method, top-left; Ice shape comparison: (Ks=0.0001), top-right; (Ks=0.0005), down-left; (Ks=0.001), down-right.

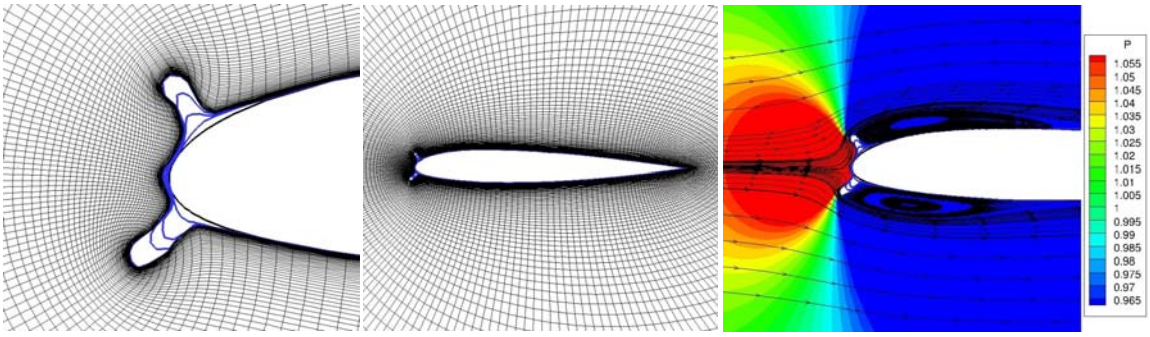

Figure 6.21: Generated grid and computed flow, (TS=5, Ks=0.001, case C16).

The collection efficiency comparison and the ice shape comparison for case C18 are shown in Figure 6.22. The generated grid and flow for a test run are shown in Figure 6.23.

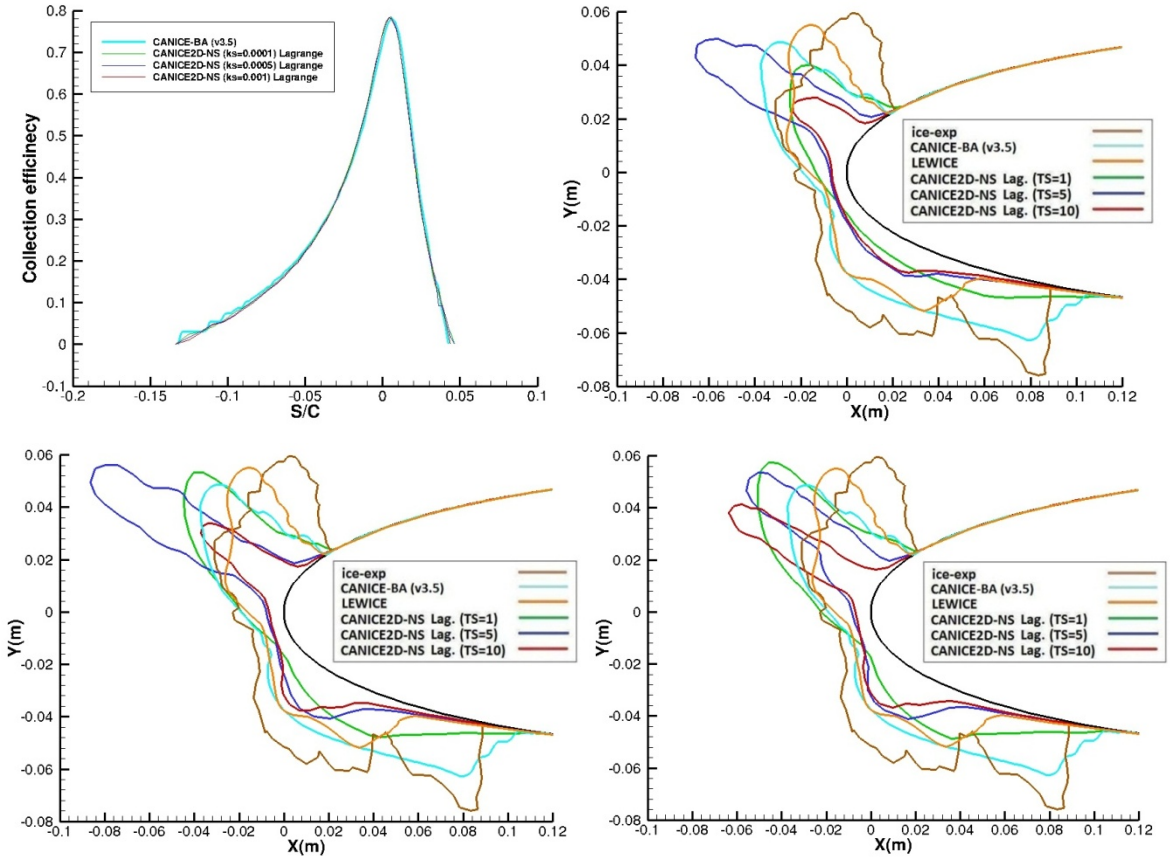

Figure 6.22: Collection efficiency comparison: Lagrangian method, top-left; Ice shape comparison: (Ks=0.0001), top-right; (Ks=0.0005), down-left; (Ks=0.001), down-right.

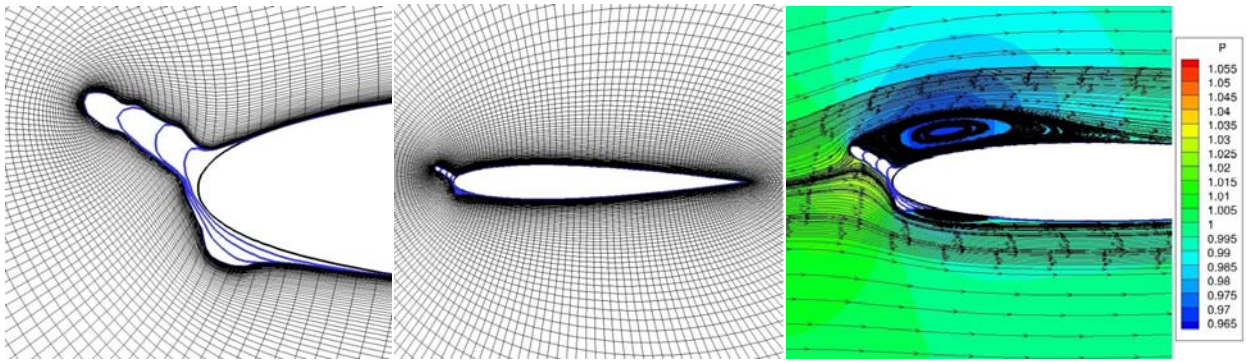

Figure 6.23: Generated grid and computed flow, (TS=5, Ks=0.001, case C18).

The collection efficiency comparison and the ice shape comparison for case C14 are shown in Figure 6.24. The generated grid and flow for a test run are shown in Figure 6.25.

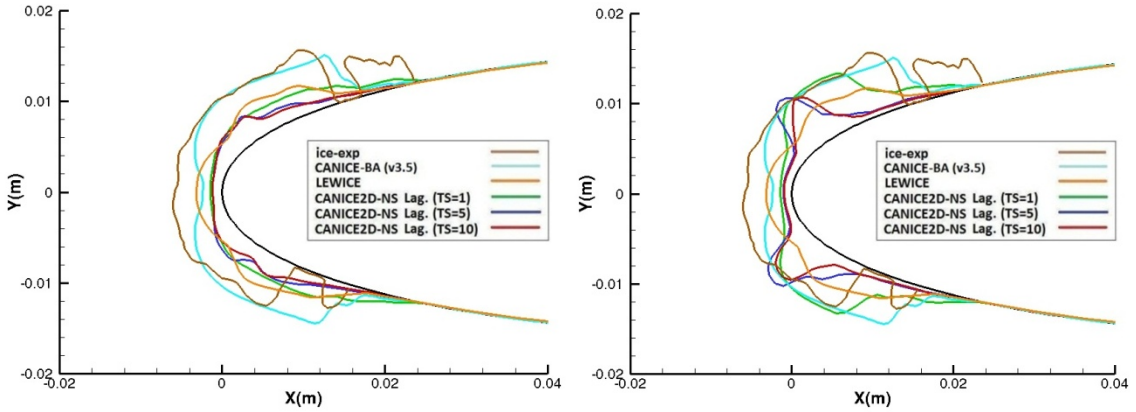

Figure 6.24: Ice shape comparison: (Ks=0.0001), left; (Ks=0.001), right.

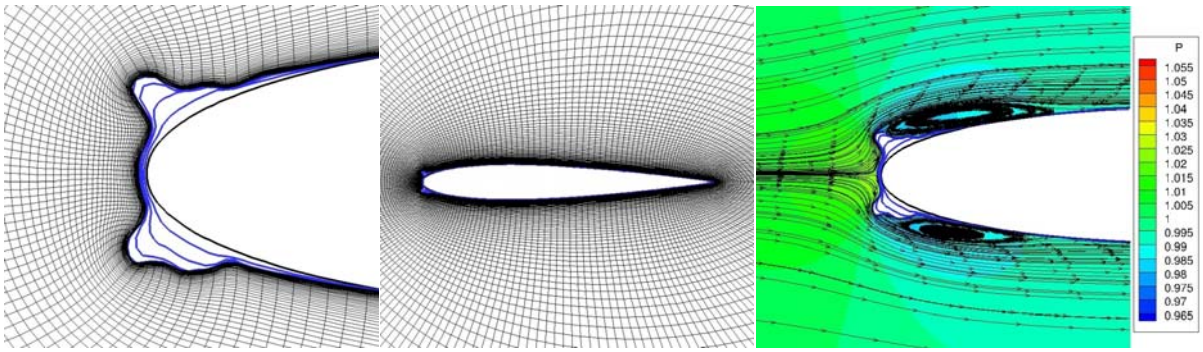

Figure 6.25: Generated grid and computed flow, (TS=5, Ks=0.001, case C14).

The collection efficiency comparison and the ice shape comparison for case C15 are shown in Figure 6.26. The generated grid and flow for a test run are shown in Figure 6.27.

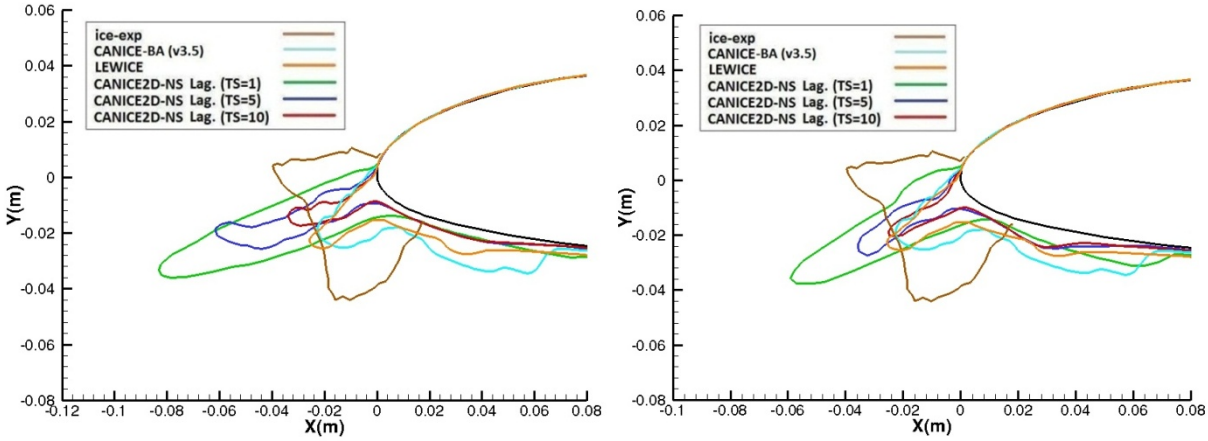

Figure 6.26: Ice shape comparison: (Ks=0.0001), left; (Ks=0.001), right.

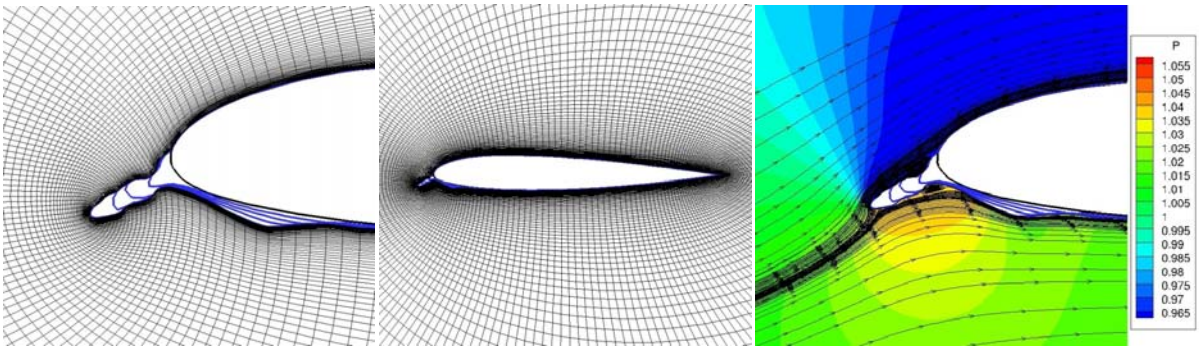

Figure 6.27: Generated grid and computed flow, (TS=5, Ks=0.001, case C15).

Figure 6.28 shows a typical unsteady flow behavior caused by complex ice horns formation in glaze ice cases.

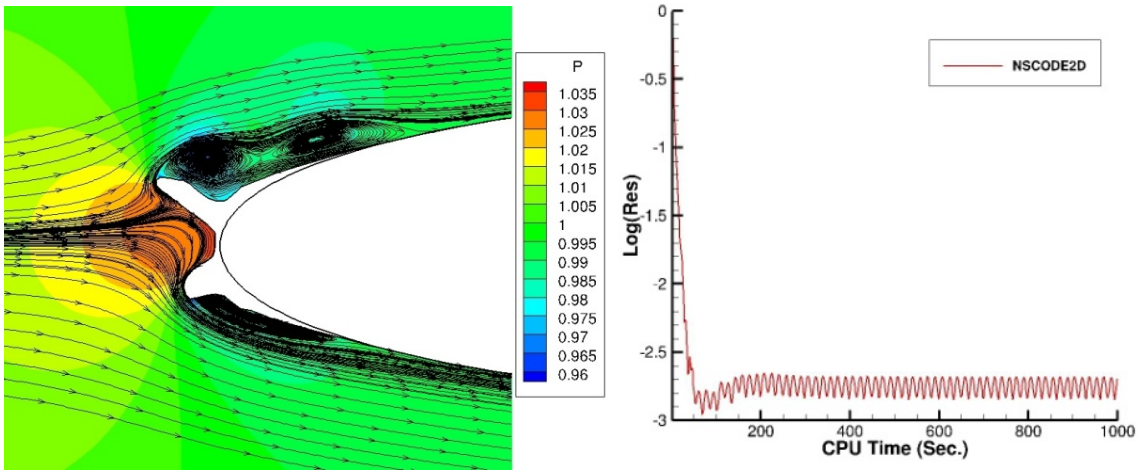

Figure 6.28: Unsteady flow: pressure contour (left), convergence (right), (case C18).

6.3 Conclusion

In total 72 individual test runs is performed for 8 icing test cases with 3 choices for number of time steps and 3 choices for roughness values. There are 6 failed test runs that are all type of complex glaze ice forms, as exemplified by Figure 6.28. The list of the 6 cases is in Table 6.2. The problems are mainly due to complex flow behavior (i.e. flow recirculation, unsteady effects) caused by ice horn growth which resulted in Lagrangian trajectory calculation failure. In general, there are 66 successful multi-steps computations which results in 91.6% success rate that here is defined as a measure of the robustness of the developed RANS based ice accretion simulation code CANICE2D-NS. Note there is no grid solver failure which shows the robustness of the developed mesh solver NSGRID2D. The computation times is also significantly reduced by the

order of 10 to around 10 minutes (for each time steps) using the upgraded framework of CANCIE2D-NS coupled with NSCODE2D and NSGRID2D compared to prior version (CANCIE2D-NS/ICEM/NSMB).

Table 6.2: NATO cases failed test runs.

NATO Cases	**TS**	**Ks**	**Crashed on Layer**
C13	10	0.0001	8
C13	10	0.0005	7
C13	10	0.001	10
C15	10	0.0005	10
C16	5	0.0005	5
C16	5	0.001	5

CHAPTER 7

GENERAL DISCUSSION

7.1 Context

Starting with a panel-method, Lagrangian based icing framework CANICE2D, a RANS based framework was created by sequential steps. First, the panel solution was replaced with the PMB3D solver. This required the introduction of a mesh generation step, done with the commercial package ICEM-CFD. Thus, a new RANS based icing framework was created, satisfying objective 1 of the research project of Section 1.4.

Then to cater for the importance of surface roughness in an icing framework, the flow solver NSMB was introduced. The work, presented in Chapter 4, allows an automated RANS based ice accretion process to reach a converged ice shape solution (or time steps independent solution) through increase in number of quasi time steps. It is shown that number of time steps (20 to 40) results in converged ice shape solution, as opposed to the widely used number of steps (3 to 5) in icing codes. Up to now, most of the ice accretion simulation codes are used based on matching the ice shape formation with the experimental data using calibration parameters such as number of time steps and roughness. Here we have shown that reaching numerically converged solution is an essential step for further improvements in ice accretion modeling.

The developed framework includes roughness analysis capabilities to further improve the ice mass formation in comparison with the experimental data, for validation purposes. An aero-icing framework has been also developed to further calculate the aerodynamic performances parameters at each steps of quasi steps ice accretion process. This allows to fully analyze the ice accretion effects and further to compute the aerodynamic stall and increase in aerodynamic drag. This work therefore addressed objective 2 of the thesis.

An automated grid generation process capable of complex ice shape mesh generation is proved to be a critical step and a bottleneck toward numerical consistency of the RANS based multi-steps aero-icing simulations. The ICEM-CFD commercial tool is shown to have significant disadvantages: automation issues, deficient grid smoothing methods, and insufficient quality grid metrics. Therefore Chapter 5 describes the development of an in-house grid generation tool, NSGRID, to address these issues. The tool is capable of automated grid regeneration with

minimal user input to generate/smooth complex ice shapes grids with improved grid metrics which are essential for RANS simulations. A novel PDE curvature based 1D surface mesh generation is developed to address ice surface mesh generation problem containing large number of concave/convex regions and sharp corners. The approach is automated with minimal user input. It captures the high curvature points and solves an elliptic equation with efficient, heuristically determined, control functions to increase mesh density in high curvature regions and decrease mesh density in between. It results in an improved mesh point distribution and reduction in field grid shock problems caused by concave regions. Note here, there is no need for surface geometry smoothing prior to surface mesh generation which is done in different mesh generation tools. This reduces the user work and preserves the real physical ice geometry. Moreover the PDE method, based on its nature, is easily extendable to 2D surface mesh generation.

In field mesh generation, NSGRID is embedded with the developed elliptic blended method combining three control functions on the right hand side with their weight factors. These source terms are RLS, SPS with algebraic parametric domain and SPS with parabolic parametric domain (named Para). The blending method with right weight factors results in a significant improvements in grid metrics for complex ice geometry mesh generation. It results in reduction in grid minimum spacing needed for RANS resolution, increase in grid orthogonaly in viscous region, reduction in grid skewness, improvements in grid stretch ratio and grid folding removal. A number of explicit and implicit solvers embedded within FMG operators are analyzed on the blended method which results in computation time reduction by about 100 times using implicit Line-SOR in I with 2 levels of FMG compared to ADI in J, Line-SOR in J (and reduction by about 65 times compared to Gauess-Seidel and Point-Jacobi).

Moreover, NSGRID is capable of multi-block smoothing with boundary updates with flexibility of different smoothing approaches application on different blocks. A variety of mesh enhancements are implemented in NSGRID such as mesh movement method, surface/domain manipulation/modifications and algebraic/parabolic grid methods, which all resulted in increase in capabilities for complex domain mesh generation.

To improve the preliminary developed RANS based multi-steps aero-icing framework of CANCE2D-NS (described in Chapter 4), two major upgrades are performed: first steps is the coupling of an efficient in-house RANS solver NSCODE2D, replacing the prior NSMB RANS

solver, and second step is the coupling of an efficient in-house grid generation solver NSGRID2D, replacing the prior ICEM-CFD mesh solver. These developments, validation and analysis are discussed in Chapter 6.

The RANS solver upgrade to NSCODE2D has improved the computation time by the order of 10 times reduction compared to prior NSMB RANS solver. It is also shown that FMG-J coarsening improves the robustness of the solver on glaze ice form flow simulation. The flow solver includes a rough wall treatment model Boeing embedded in S-A turbulence model for ice roughness analysis. Moreover NSCODE is capable of Chimera method which facilities the RANS aero-icing simulation for high-lift configurations (future work).

The new upgraded CANICE2D-NS framework is further used to analyze a variety of standard case studies to evaluate the robustness of the multi-steps RANS based ice accretion process. Here robustness is considered based on the number of successful ice accretion computations (66 run) among large number of multi-steps test runs (72 individual run), which represent 91.6% success. Note that the flow, grid and ice solvers properties are configured in a sense to relax the computations to have minimum failure. The failed test cases are mainly due to complex flow behavior (i.e. flow recirculation, unsteady effects) which resulted in Lagrangian droplet computation failure. An important observation is that there is no grid solver failure using NSGRID2D mesh solver that proves the robustness of the grid solver for complex ice shape mesh generation. Note that both NSCODE2D flow solver and NSGRID2D grid solver have significantly improved the multi-steps ice accretion computation time by order of over 10 times compared to preliminary framework (NSMB/ICEM-CFD).

In general, these developments form the baseline/benchmark RANS multi-steps ice accretion simulations at Polytechnique de Montreal, for further study of ice accretion physical modeling and extension to 3 dimensional domains.

7.2 Limitations

The RANS based multi-steps ice accretion includes number of limitations in each step of grid, flow and ice solvers.

Grid solver is shown to be a critical bottleneck in multi-steps aero-icing simulation. ICEM-CFD mesh solver showed number of limitations for complex ice form problems such as grid smoothing deficiencies, inefficient grid metrics and automation restrictions.

The newly developed grid generation code NSGRID demonstrated significant capabilities to address all the issues encountered with the ICEM-CFD grid solver. A limitation of the grid solver comes when, within a multi-steps ice formation, closing cavities occur which require surface treatments. Further testing on more complex geometries, including multi-element airfoil, could reveal more limitations.

One of the major limitations in steady state RANS solver application is due to unsteady flow behavior caused by complex ice shapes. This can result in failure in quasi-steady ice accretion process using a Lagrangian approach. Therefore there is a need for further investigation and development toward fully unsteady flow and ice accretion simulation. In this work, for complex case studies, the steady state RANS solver properties is defined in a manner to dissipate the problem as much as possible using the JST dissipation factors, CFL values, and multigrid algorithm.

Computation of convective heat transfer (for ice accretion thermodynamics calculation) is performed using the RANS skin friction solution and semi-empirical roughness based turbulence heat transfer formula that is another limiting issue. This results in low heat transfer and consequently low ice mass accumulation in the laminar region of stagnation points. Therefore there is a need to study convective heat transfer computation directly through RANS modeling.

A limiting issue in ice accretion simulation is the conservation of ice mass after geometry update. There is a need to study this problem to preserve the amount of accumulated ice mass at each time steps. This issue results in low ice mass solution results in converged ice shape (20-40 steps) compared to experimental data. Our literature survey on this particular issue, not examined in this work, indicate that the mass conservation is not enforced in many frameworks i.e. ONERA-2D. The other causing factor also can be the surface roughness value that may change in time and space and affects the convective heat transfer values and consequently ice mass growth. Therefore there is a need to fully study the uniform/non-uniform roughness modeling and its effects.

The other limiting issue in ice accretion simulation is the assumption of single stagnation point, simple water runback modeling and traditional Messinger model calculation that are performed in this work. This results in error in ice mass growth computation as the outcome of wrong water runback direction assumption. Therefore there is a need for further study in

modeling of multi-stagnation points problem and water runback computations toward application of iterative Messinger method or SWIM.

Lagrangian droplet computation is another limiting issue in traditional ice accretion simulation. As complexity of ice increases, it results in more complex flow behavior and consequently unsteady flow behavior. This issue in addition to RANS flow interpolation method, mesh size and the Lagrangian method properties such as time increment, releasing point increment and positions, can affect the droplet computation and results in computation failure and droplets loss. Therefore there is a need to study the Eulerian method application that is directly solved using a PDE approach and can impact the numerical solutions.

CHAPTER 8

CONCLUSION AND RECOMMENDATION

8.1 Conclusion

A step by step approach is used to develop a novel, aerodynamic ice accretion and ice degradation effect framework based on a RANS formulation. The work has addressed all three specific objectives of the thesis. Chapter 4 and 6 address the first and second specific objectives to examine the impact of RANS flow solver on ice accretion framework and examine multi-steps ice accretion shape convergence. Chapter 5 addresses the third specific objective to develop novel grid generation algorithms specifically for ice shape/growth.

The work starts with Polytechnique Montreal's CANICE framework, which contains a panel solver, a boundary layer code and an ice accretion suite based on the Messinger model. Two 3D RANS solver, PMB and NSMB are coupled to CANICE and used for 2D airfoils. The importance of having a turbulence model handling rough surfaces in ice accretion validation is demonstrated. The RANS codes are using meshes created with the commercial ICEM-CFD grid generation package. A multi-steps icing capability is developed, thanks to using a mesh regeneration procedure instead of using a mesh movement approach. The fully automated procedure allows providing solutions for over 200 multi-steps calculations via a quasi-steady approach. The results indicate the need to perform 20-40 steps to reach engineering converged results, as opposed to the 1-5 steps procedure used in other studies.

The meshes created with ICEM-CFD have a few deficiencies, notably that they are expensive to run, need high expertise to generate them, have an inefficient replay function but most importantly, generated meshes with poor grid metrics. A new 2D mesh generation code, NSGRID, is created to address these issues. A novel curvature-based surface adaption algorithm is developed to address the grid shock problem in concave regions by automatically refine/coarsen in concave/convex regions, respectively. The final mesh is created by a PDE approach, effective and comprehensive. An elliptic Poisson solver with carefully devised blended source terms enables to attain excellent grid metrics in term of orthogonality, skewness and stretching ratios. Furthermore, a detailed study on the selection of an effective Poisson solver smoother in the content of a Full Multigrid operator allows generating meshes in 5 seconds for a

4 order residual drop, and to machine accuracy if needed. Not only does NSGRID generate better grids, but it also allows the RANS solvers to assemble a much better conditioned system.

Two major improvements are performed to the preliminary framework of CANICE2D-NS coupled with NSMB flow solver and ICEM-CFD grid solver. First step, replacing NSMB with an efficient 2D RANS solver NSCODE, to significantly improve the multi-steps computation time and FMG robustness problem, using NSCODE FMG-J coarsening operator. Second step, replacing ICEM-CFD with an efficient in-house mesh solver NSGRID2D, to significantly improve the remeshing process as detailed in Chapter 5. Finally the new framework is validated on standard NATO cases including 72 individual test runs. The success rate of the calculation is 91.6%, which is considered as a robustness criteria of the CANICE2D-NS framework. The 6 failed computations are mainly due to the complex flow behavior caused by glaze ice horn growth which is inevitable. The grid solver NSGRID has shown significant robustness with no failure for these test runs. Moreover the new framework improved the computation time by over 10 times compared to the prior framework (ICEM/NSMB).

A 3D extension of NSGRID is successfully performed, and is coupled to NSMB-ICE, a 3D chimera/level-set/Eulerian icing, framework developed via collaboration with University of Strasbourg. The work enables 2D and 3D ice accretion and icing effects analysis within a multi-step procedure. This work is excluded from the thesis, but demonstrates the feasibility of the proposed algorithms for 3D problems.

8.2 Future work

The following steps are proposed for future research and improvements in mesh generation and ice accretion simulations:

- Full extension of blended grid generation method to 3 dimensional meshes, with quantitative analysis.
- Additional source terms implementation in the blended method.
- Optimization of the grid factors and parameters in the blended method for grid quality and convergence improvements.
- Extension of the 1D PDE curve based mesh generation to 2D surface mesh generation for 3D problems.

- Full extension of newly developed multi-time steps RANS based ice accretion codes, CANICE-NS and NSCODE-ICE to 3D.
- Investigation of unsteady effects in icing simulations.

Other recommendations, indirectly addressed in the thesis, are:

- Further physical studies on the impact of non-uniform roughness, Supercooled Large Droplet model.
- Detailed studies of the various thermodynamics models such as Iterative Messinger, Shallow-Water Icing Model on the final ice shapes.
- A study of more advanced heat transfer coefficient models.

REFERENCES

[1] National Transportation Safety Board Aircraft Accident Report, "In Flight Icing Encounter and Uncontrolled Collision with Terrain, Comair Flight 3272," *Embraer EMB-120RT,* N265CA Monroe, Michigan, January 9, 1997.

[2] Brahimi, P. M.T. Tran and Paraschivoiu. I. "Numerical Simulation and Thermodynamic Analysis of Ice Accretion on Aircraft Wings," Technical Report, *Bombardier Aerospace Group*, Quebec, Canada, May, 1994.

[3] "Aircraft Icing Handbook," *Civil Aviation Authority*, New Zealand, 2000.

[4] Broeren A. P. and Bragg M. B., "Flowfield measurements about an airfoil with leading edge ice shapes," *Journal of Aircraft*, Vol. 43, No. 4, 2006.

[5] Saeed F., "State of the Art Aircraft Icing and Anti-icing Simulation," *25th Annual ARA Congress,* Cleveland, Jul. 2000.

[6] Chi X., Zhu B. and Shih T. I. P., "Computing aerodynamic performance of 2d iced airfoil: Blocking topology and grid generation," *40th Aerospace Science Meeting & Exhibit*, 2002.

[7] Addy H. E., "Ice Accretions and Icing effects for Modern Airfoils," Technical Report, NASA, TP-2000-210031, Apr. 2000.

[8] Wright W. B., "Users Manual for the NASA Glenn Ice Accretion Code LEWICE Version 2.0," NASA CR-209409, Sep. 1999.

[9] Gouttebroze S., Saeed F. and Paraschivoiu I., "CANICE-Capabilities and Current Status," *NATO/RTO Workshop, Assessment of Icing Code Prediction Capabilities*, CIRA, Italy, Dec. 2000.

[10] Wright W., Gent R. W. and Guffond D., "DRA/NASA/ONERA Collaboration on Icing Research, Part II-Prediction of Airfoil Ice Accretion," NASA CR-202349, May 1997.

[11] Messinger B. L., "Equilibrium Temperature of an Unheated Icing Surface as a Function of Airspeed," *Journal of Aeronautical Sciences*, Vol. 20, No. 1, Jan. 1953, pp. 29, 42.

[12] Thompson D. and Soni B., "ICEG2D-An integrated Software Package for Automated Prediction of Flow Fields for Single-Element Airfoils with Ice Accretion," NASA CR-209914, Feb. 2000.

[13] Hasanzadeh, K., and Laurendeau, E., "Validation and User Manual of CANICE2D-NS" Technical Report, Bombardier Aerospace, Oct. 2014.

[14] Beaugendre H., Morency F. and Habashi W., "Development of A Second Generation In-flight Icing Simulation Code," *Journal of Fluid Engineering*, Vol. 128, No. 2, 2006, pp. 378, 387.

[15] Brahimi M. T., Tran P. and Paraschivoiu I., "Prediction of Rime Ice Shapes on Wings," *14th Canadian Congress of Applied Mechanics*, CANCAM 93, Kingston, Canada, May 1993.

[16] Saeed F., Morency F. and Paraschivoiu I., "Numerical Simulation of a Hot-Air Anti-icing System," *38th AIAA Aerospace Sciences Meeting,* Reno, Nevada, Jan. 2000, AIAA paper 2000-0630.

[17] Tran P., Brahimi M. T., Paraschivoiu I., Pueyo A. and Tezok F., "Ice Accretion on Aircraft Wings with Thermodynamic Effects," *AIAA Journal of Aircraft*, Vol. 32, No. 2, Mar. 1995, pp. 444, 446,.

[18] Tran P., Brahimi M. T., Sankar L.N, and Paraschivoiu I., "Ice Accretion Prediction on Single and Multiple-Element Airfoils and the Resulting Performance Degradation," *35th AIAA Aerospace Sciences Meeting,* Reno, Nevada, Jan. 1997.

[19] Pueyo, A, et al., "Comparisons Exercises of Ice Accretion Simulations with 2D and 3D solvers, SAE Aircraft and Engine Icing International Conference, Seville, Spain, Sept. 2007, SAE paper 2007-01-3338.

[20] Hasanzadeh K., Laurendeau E., and Paraschivoiu I., "Quasi-steady convergence of multi-step Navier-Stokes icing simulations," Journal of Aircraft, June 2013, DOI: 10.2514/1.C032197.

[21] Steger J. L., and Rizk Y., M., "Generation of Three Dimensional Body-Fitted Grids by Solving Hyperbolic Partial Differential Equations, Technical Report, NASA, TM-1985-86753, June 1985.

[22] Spekreijse S. P., and Boerstoel J. W., "Multiblock Grid Generation – Part I: Elliptic grid generation methods for structured grids," 27th Computational Fluid Dynamics Course, Belgium, March 1996.

[23] Thompson J., F., Soni B., K., and Weatherill N., P., "Handbook of Grid Generation," CRC Press, 1999.

[24] Cebeci T., Shao J.P., Kafyeke F., and Laurendeau E., "Computational Fluid Dynamics for Engineering," Springer, New York, 2005.

[25] Vickerman M., et al., “SmaggIce: Surface Modeling and Grid Generation for Iced Airfoils: Phase 1 Results,” AIAA 2000-0235, 2000.

[26] Vos J.B. and Rey B., “Implementation of roughness modeling in the NSMB CFD Code,” *CFS Engineering*, Technical Report, 2007.

[27] Aupoix B. and Spalart P. R., “Extension of the Spalart-Allmaras turbulence model to account for wall roughness,” *Heat and Fluid Flow Journal*, Vol. 24, 2003, pp. 454, 46.

[28] Hellsten A. and Laine S., “Extension of the k-ω-SST turbulence Model For Flows Over Rough Surface,” AIAA Atmospheric Flight Mechanics Conference, New Orleans, Louisiana, 1997.

[29] Croce G., Candido E., Habashi W., et al., “FENSAP-ICE: Analytical Model for Spatial and Temporal Evolution of In-Flight Icing Roughness,” *Journal of Aircraft*, 47(4), August 2010.

[30] Mavriplis D. J., “Multigrid Solution of the Discrete Adjoint for Optimization Problems on Unstructured Meshes,” *AIAA Journal*, Vol. 44, No. 1, Jan. 2006, pp. 42, 50.

[31] Hasanzadeh K., Mosahebi A., Laurendeau E., and Paraschivoiu I., “Validation and Verification of Multi-Steps Icing Calculation Using CANICE2D-NS Code," AIAA Fluid Dynamics and Co-located Conferences and Exhibit, San Diego, California, USA, June 2013.

[32] Pena, D., Hoarau, Y., Laurendeau E., “Parallel computing of icing on three-dimensional airfoils,” 11th World Congress on Computational Mechanics (WCCM XI), 5th European Conference on Computational Mechanics (ECCM V), 6th European Conference on Computational Fluid Dynamics (ECFD VI), July 20 - 25, 2014, Barcelona, Spain.

[33] Beaugendre Y., et al., “A Finite Element Method Study of Eulerian Droplets Impingement Models,” *International Journal for Numerical Method in Fluids*, 29, 429-449, 1999.

[34] Bourgault-Côté, S., Laurendeau, E., “Two-Dimensional/Infinite Swept Wing Ice Accretion Model,” *AIAA (SciTech 2015),* 5-9 January, Kissimmee, FL, USA.

[35] Bourgault Y., Beaugendre H., Habashi W., “Development of a Shallow-Water icing Model in FENSAP-ICE,” *Journal of Aircraft*, 37, 4, July 2000.

[36] Aliaga C., Aube M., Barruzi G., Habashi W., Nadarajah S., “A Third-generation In-flight icing Code: FENSAP-ICE-Unsteady,” *SAE Conference*, 2007.

[37] Beaugendre H., Morency F., Habashi W., “FENSAP-ICE: Roughness Effects on Ice Shape Prediction,” *41st Aerospace Science Meeting & Exhibit, Reno*, Nevada, Jan 2003.

[38] Beaugendre H., Morency F., Habashi W., “ICE3D FENSAP-ICE’s 3D In-Flight Ice Accretion Module,” *40th Aerospace Sciences Meeting & Exhibit*, Nevada, Jan 2002.

[39] Hasanzadeh K., Laurendeau E., Saeed F., and Paraschivoiu F., “Wing Aerodynamic Performance Analysis and Stall Prediction Using CANICE2D-NS Icing Code," *20th Annual Conference of the CFD Society of Canada,* Canmore, AB, Canada, May 2012.

[40] Chung J., Choo Y., Reehorst A., Potapczuk M., Slater J., “Navier-Stokes analysis of the flowfield characteristics of an ice contaminated aircraft wing," *Technical Report TM-1999-208897*, NACA, Jan. 1999.

[41] Hasanzadeh, K., Laurendeau, E., and Paraschivoiu, I., “Adaptive curvature control grid generation algorithms for complex glaze ice shapes RANS simulations,” *AIAA (SciTech 2015)*, 5-9 January, Kissimmee, FL, USA.

[42] ANSYS ICEM CFD 11.0, *Tutorial Manual*, Jan. 2007.

[43] Badcock K.J., Richards B.E., Woodgate M.A., “Elements of Computational Fluid Dynamics on block structured grids using implicit solvers,” *Progress in Aerospace Sci*., vol 36, pp 351-392, 2000.

[44] Vos, J.B. et al., “NSMB Handbook 4.5_Update for 6.0,” *Report, EPFL University*, June 2012.

[45] Levesque, A.T., Pigeon , P., Deloze, T., Laurendeau, E., “An overset grid 2D/Infinite swept wing URANS solver using Recursive Cartesian Virtual Grid method” *AIAA (SciTech 2015)*, 5-9 January, Kissimmee, FL, USA.

[46] Hasanzadeh K., Bourgault-Cote S., Laurendeau E., Brette C., and Paraschivoiu I., “Validation of multi-time steps Lagrangian/Eulerian RANS based icing code CANICE2D-NS,” *Canadian Aeronautics and Space Institute Conference (CASI)*, Montreal, Canada, 2015.

[47] Hasanzadeh K., Pena D., Hoarau Y., and Laurendeau E, “Multi-steps icing calculations using a 3D multi-block structured mesh generation procedure,” *SAE International Conference on icing of Aircraft,* Prague, Czech Republic, 2015.

[48] Wright W. B., Rutkowski A., “Validation Results for LEWICE 2.0,” NASA CR-208690, Jan. 1999.

[49] Saeed F., "A Three-Dimensional Water Droplet *Trajectory and Impingement Analysis Program," 23rd* AIAA Applied Aerodynamic Conference, Toronto, Ontario, Canada, June 2005.

[50] Brette C., "Three Dimensional Water Droplet Impingement Simulation on Aircraft," *Technical Report*, École Polytechnique de Montréal, May-August 2003.

[51] Saeed F., Gouttebroze S., and Paraschivoiu I., "Modified CANICE for Improved Prediction of Airfoil Ice Accretion," *8th Aerodynamic Symposium of the CASI 48th Annual Conference,* Toronto, Canada, pp. 283-289, Apr. 29-May 2, 2001.

[52] Kreeger R.E. and Wright W.B. "The Influence of Viscous Effects on Ice Accretion Prediction and Airfoil Performance Predictions," *Technical Report*, TM-2005-213593, NACA, March 2005.

[53] Bertin J., "Aerodynamic for Engineering, Prentice hall, Fourth Edition," New Jersey, 2002.

[54] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," *NASA/CR-2014-218178,* March 2014.

[55] Sorenson L., "A Computer Program to Generate Two Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equation," *Technical Report*, NASA, May 1980.

[56] Piperni P., "A fundamental approach to the problem of domain decomposition in structured grid generation," *Ecole Polytechnique de Montreal*, PhD thesis, 2003.

[57] Blazek J., "Computational Fluid Dynamics: Principles And Applications," Elsevier, New York, 2001.

[58] Thompson D., Soni B., "ICEG2D-A Software Package for ice Accretion Prediction," *Technical Report*, NASA, June 2002.

[59] Koomullil R.P., Thompson D.S. and Soni B.K. "Iced airfoil simulation using generalized grids," *Applied Numerical Mathematics*, 46, 319-330, 2003.

[60] Hicken J. E. and Zingg D. W., "Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement," *AIAA Journal*, Vol. 48, No. 2, Feb. 2010.

[61] Truong A. H., Oldfield C. A. and Zingg D. W., "Mesh Movement for a Discrete-Adjoint Newton-Krylov Algorithm for Aerodynamic Optimization," *AIAA Journal*, Vol. 46, No. 7, Jul. 2008.

[62] Ozcer I., et al., "FENSAP-ICE: Numerical Prediction of Ice Roughness Evolution and its Effects on Ice Shape," *SAE Aircraft & Engine Icing International Conference*, Chicago, Illinois, USA, June 2011.

[63] Hasanzadeh k., Mosahebi A., Laurendeau E., and Paraschivoiu I., "Framework for Multi-Steps Icing Simulation Code CANICE2D-NS," *Canadian Aeronautics and Space Institute Conference (CASI)*, Toronto, Canada, April– May, 2013.

[64] Serkan O., Erhan T., Murat C., "Parallel Computing Applied to Three-Dimensional Droplet Simulation in Lagrangian Approach," *SAE, Chicago*, Illinois, USA, June 2011.

[65] Wirogo S., Srirambhatla S., "An Eulerian Method to Calculate The Collection Efficiency on Two and Three Dimensional Bodies," *41st Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2003.

[66] Hospers J., Hoeijmakers H., "Eulerian method for Ice accretion on Multi-Element Airfoil Sections," *48th AIAA Aerospace Sciences Meeting*, Orlando, Florida, January 2010.

[67] Brahimi M. T., Tran P., Chocron D., Tezok F. and Paraschivoiu I., "Effects of Supercooled Large Droplets on Ice Accretion Characteristics," *35th AIAA Aerospace Sciences Meeting,* Reno, Nevada, Jan. 1997.

[68] Bragg M., Loth E., "Effects of Large-Droplet Ice Accretion on Airfoil and Wing Aerodynamics and Control," *Technical Report*, Office of Aviation Research, Washington, March 2000.

[69] Peng K., Xinxin W., "Super-cooled Droplets consideration in the droplet impingement simulation for aircraft icing," *Procedia Engineering*, 17, 151-159, 2011.

[70] Meyers T., "Extension of the Messinger Model for Aircraft Icing," *AIAA Journal*, 39, 2, Feb 2001.

[71] Paraschivoiu I., Tran P., and Brahimi M.T., "Prediction of the Ice Accretion with Viscous Effects on Aircraft Wings," *AIAA Journal of Aircraft*, Vol. 31, No. 4, pp. 855-861, Jul.-Aug. 1994.

[72] Widhalm M., Ronzheimer A., Meyer J., "Lagrangian Particle Tracking on Large Unstructured Three-Dimensional Meshes," *46th AIAA Aerospace Sciences Meeting and Exhibit* , Reno, Nevada, U.S.A., 2008.

[73] Thomas, P. D., and Middlecoff, J. F., "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations", *AIAA Journal*, Vol. 18, 1979, pp. 652-656.

[74] Sorenson, R. L., and Steger, J. L., "Grid Generation in Three Dimensions by Poisson Equations with Control of Cell Size and Skewness at Boundary Surfaces", *Advances in Grid Generation,* FED-Vol. 5, Ed*. K.N. Ghia and U. Ghia, ASME Applied Mechanics, Bioengineering, and Fluids Engineering Conference*, Houston, 1983.

[75] Steinbrenner J., and Chawner J., "GRIDGEN's Implementation of Partial Differential Equation Based Structured Grid Generation Methods," *Proceedings, 8th International Meshing Roundtable*, South Lake Tahoe, CA, U.S.A., October 1999, pp. 143-152.

[76] Vickerman M., et al., "Toward an Efficient Icing CFD Process Using and Interactive Software Toolkit-SmaggIce 2D," *4th Aerospace Science meeting and Exhibit*, Nevada, Jan 2002.

[77] Chung J., et al., "Effects of Airfoil Ice Shape Smoothing on the Aerodynamic performance," *AIAA Paper*, 98, 3242, July 1998.

[78] Choo Y., et al., "User Manual for beta Version of Turbo-Grid, A Software System for interactive Two-Dimensional Boundary/Field Grid Generation, Modification, and Refinement," *Technical Report*, NASA, TM-1998-206631, Oct. 1998.

[79] Thompson D., Soni B., "Semi-structured Grid Generation in Three Dimensions Using A Parabolic Marching Scheme," *AIAA Journal*, 40, 2, pp. 391-393, 2000.

[80] Thompson D., Soni B., "Generation of Quad- and Hex-Dominant Semi-structured Grids using an Advance Layer Scheme," *8th International Meshing Roundtable Proceeding*, 1999.

[81] Chalasani S., Thompson D., "Quality improvements in extruded meshes using topologically adaptive generalized elements," *International Journal for Numerical Methods in Engineering*, 60,1139-1159, 2004.

[82] Sorenson R. L., "Elliptic Generation of Composite Three-Dimensional Grids about Realistic Aircraft," *Technical Report*, NASA, TM-1986-88240, March 1986.

[83] Steger J. L., and Sorenson, R. L., "Automatic mesh-Point Clustering near a Boundary in Grid Generation with Elliptic Partial Differential Equations," *Journal Of Computational Physics,* Vol. 33, 1979, pp. 405-410.

[84] Spekreijse S., P., "Elliptic Grid Generation Based on Laplace Equations and Algebraic Transformations," *Journal of Computational Physics*, 118, pp. 38-61, 1995.

[85] Collis S. S., "An Introduction to Numerical Analysis for Computational Fluid Mechanics," *Technical Report*, Sandia National Laboratories, California, April 2005.

[86] Spitaleri R. M., "Full-FAS multigrid grid generation algorithms," *Applied Numerical Mathematics,* Vol. 32, 2000, pp. 483-494.

[87] Briggs W. L., Henson, V. E., and McCormick, S. F., "A Multigrid Tutorial," *Society for Industrial and Applied Mechanics*, 2nd Edition, 2000.

[88] Badcock K.J., "Evaluation of Results from a Reynolds Averaged Multiblock Code Against F-16XL Flight Data," *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, AIAA Paper 2007-0490, Jan. 2007.

[89] Feszty D., Badcock K.J., and Richards B.E., "Utilising CFD in the investigation of high speed unsteady spiked body flows," *Aeronautical Journal*, pp 161-174, April 2002.

[90] Marques S., Badcock K.J., Gooden H.M., Gate S., and Maybury W., "Validation Study For Prediction of Iced Aerofoil Aerodynamics," *Aeronautical Journal*, 114-1152, February 2010.

[91] Vos J., et al., "NSMB 6.04 User Guide, *CFS Engineering*," Technical Report, Sep. 2011.

[92] Deloze T., Hoarau Y. and Dusek J., "Transition scenario of a sphere freely falling in a vertical tube," *Journal of Fluid Mechanics*, in press, 2012.

[93] Hoarau Y., Vos J., Deloze T., Charbonnier D. and Rey B., "Automatic Chimera Method for Moving Bodies in NSMB," *Overset Grid Symposium*, Dayton, Ohio, Oct. 2012.

[94] Pigeon A., Levesque A. T., and Laurendeau E., "Two-Dimensional Navier-Stokes Flow Solver Developed at Ecole Polytechnique de Montreal," CFD Society of Canada, 22nd annual conference, CFDSC, 2014.

[95] Kays W. M., and Crawford M. E., "Convective Heat and Mass Transfer," *3rd edition, McGrew-Hill*, 1993.

[96] Hasanzadeh K., Laurendeau E., and Paraschivoiu I., "Adaptive curvature control grid generation algorithms for complex glaze ice shapes RANS simulations," AIAA Journal, Nov. 2014, manuscript ID: 2014-11-J054076. (Accepted)

[97] Vos J., Rizzi A., Darracq D. and Hirschel E., "Navier-Stokes solvers in European aircraft design," *Progress in Aerospace Sciences*, Vol. 38, 2002, pp. 601, 697.

[98] Vos J., Rizzi A., Corjon A., Chaput E. and Soinne E., "Recent Advances in aerodynamics inside the NSMB (Navier-Stokes Multiblock) Consortium," *AIAA Journal*, 1998.

[99] Tinoco, E., et al., "Summary of the Fourth AIAA CFD Drag Prediction Workshop," *28th AIAA Applied Aerodynamics Conference*, Chicago, Illinois, June 2010.

[100] Rumsey C., Long M., Stuever B. and Wayman T., "Summary of the First AIAA CFD High Lift Prediction Workshop," *49th AIAA Aerospace Sciences Meeting*, Orlando, Florida, Jan. 2011.

[101] Schuster D. M., Chwalowski P., Heeg J. and Wieseman C. D., "A Summary of Data and Findings from the First Aeroelastic Prediction Workshop," *7th International Conference on Computational Fluid Dynamics (ICCFD7)*, Big Island, Hawaii, Jul. 2012.

[102] Braza M. et al., "Feedback effects and stochastic forcing response of the transonic buffet around an airfoil including trailing-edge-plate at high Reynolds," *6th European Congress on Computational methods in Applied Sciences and Engineering*, Vienna, Austria, Sept. 2012.

[103] Abbott I. H. and Doenhoff A. E. V., "Theory of wing sections, including a summary of airfoil data," *Dover Publications*, New York, 1959.

[104] Cook P. H., McDonald M. A. and Firmin M. C. P., "Aerofoil RAE 2822 - Pressure Distributions, and Boundary Layer and Wake Measurements," *Experimental Data Base for Computer Program Assessment*, AGARD Report AR 138, 1979.

[105] Rao D. M. and Huffman J. K., "Hinged strakes for enhanced manoeuvrability at high angles of attack," *Journal of Aircraft*, Vol. 19, No. 4, 1980.

[106] Chi X., et al., "CFD Analysis of the Aerodynamics of a Business-Jet Airfoil with Leading-Edge Ice Accretion," *42nd Aerospace Science Meeting and Exhibit*, AIAA, Nevada, Jan 2004.

[107] Addy H. E., et al., "Effects of High-Fidelity Ice-Accretion Simulations on Full-Scale Airfoil Performance," *Journal of Aircraft*, Vol. 47, No. 1, Jan.-Feb. 2010.

[108] Potapczuk M. G., "A Review of NSA Lewis Development Plans for Computational Simulation of Aircraft icing," *Technical Report*, NASA, TM-1999-208904, Jan. 1999.

[109] Bragg M. B., Broeren A.P., Blumenthal L. A., "Iced-airfoil aerodynamics," *Progress in Aerospace Sciences*, Elsevier, Vol. 41, 2005, pp. 323-362.

[110] Kreeger R., et al., "SmaggIce 2.0: Additional Capabilities for Interactive Grid Generation of Iced Airfoils," *45th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, Jan 2007.

[111] Dafa'Alla A. A., "Ice Effect on the Aerodynamic Characteristics of Aerofoils," *20th AIAA Applied Aerodynamics Conference*, St. Louis, Missouri, June 2002.

[112] Thompson J. F., "A General Three Dimensional Elliptic Grid Generation System on a Composite Block Structure," *Computational Methods in Applied Mechanics and Engineering*, Vol. 64, 1987, pp. 337-411.

[113] Spitaleri R. M., and Micacchi, V., "A multiblock multigrid grid generation method for complex simulations," *Mathematics and Computers in Simulation*, Vol. 46, 1998, pp. 1-12.

[114] Tecplot.360 2013, User's Manual, Release 1, 2013.

[115] Thompson D. S., and Soni B. K., "ICEG2D(v2.0)-An integrated Software Package for Automated Prediction of Flow Fields for Single-Element Airfoils with Ice Accretion," *Technical Report*, NASA, CR-2001-210965, June 2001.

[116] Chung J. J., and Addy E. H.., "A numerical Evaluation of Icing Effects on a Natural Laminar Flow Airfoil," *Technical Report*, NASA, TM-2000-209775, Jan. 2000.

[117] Jameson A., Schmidt S., and Turkel E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," *AIAA Paper 81-1259, AIAA 14th Fluid and Plasma Dynamic Conference*, Palo Alto, June 1981.

[118] Pierce, N.A. and Giles B., "Preconditioned Multigrid Methods for Compressible Flow Calculations on Structured Meshes," *Journal of Computational Physics*, Vol. 136, 1997, pp. 425-445.

[119] Pueyo A., Chocron D. and Kafyeke F., "Improvements to the Ice Accretion Code CANICE," *9th Canadian Aeronautics and Space Institute Aeronautics Conference*, Toronto, May 2001.

APPENDIX A – ADDITIONAL METHODOLOGIES DESCRIPTION

Content:

1. **Roughness effect implementation within turbulence model**
2. **Grid Metrics**

1. Roughness effect implementation within turbulence model

Surface roughness is one of the main parameters influencing the surface skin friction and heat transfer, and has a major influence in the ice accretion process. Roughness values are usually assumed to be constant on the surface, but in reality, ice shape roughness varies in space and time [1-3]

Rough wall treatment implementation in Spalart-Allmaras and k-ω models are examined here. There are two extensions for the Spalart-Allmaras roughness implementation: ONERA and Boeing [4]. The assumption is that the roughness size in any direction is smaller than the boundary layer thickness. The roughness is computed using sand grain roughness height model [5]. The effects of roughness are incorporated in the S-A model by affecting the turbulent eddy viscosity in the wall area. One can relate the roughness height to the changes in velocity profile that changes the wall skin friction.

The ONERA extension of the Spalart-Allmaras model is applied by defining non-zero value for turbulent viscosity at the wall, achieved by improving the value of transport quantity $\tilde{\nu}$ at the wall. To determine the transport quantity at the wall, the dimensionless form of the Spalart-Allmaras transport equation and momentum equation with viscosity ν and friction velocity u_τ change to (respectively) [4, 6]:

$$0 = c_{b1}\tilde{S}^{+}\tilde{\nu}^{+} - c_{w1}f_w\left(\frac{\tilde{\nu}^{+}}{d^{+}}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial y^{+}}\left(\tilde{\nu}^{+}\frac{\partial\tilde{\nu}^{+}}{\partial y^{+}}\right) + c_{b2}\left(\frac{\partial\tilde{\nu}^{+}}{\partial y^{+}}\right)^2\right] \quad (1)$$

$$\frac{\partial u^{+}}{\partial y^{+}} - < u'v' > = (1 + \nu_t^{+})\frac{\partial u^{+}}{\partial y^{+}} = 1 \quad (2)$$

By imposing the boundary conditions $\tilde{\nu}^{+}$at the wall and into the logarithmic region, and solving the addressed non-dimension transport equation, the $\tilde{\nu}$ and $\tilde{S}$ solutions are obtained. The velocity profile included in $\tilde{S}$ is used to determine the velocity shift Δu^{+}. Changes in velocity profile is thus related directly to the skin friction value on the wall. The wall distance is also shifted to

control the imposed wall boundary value via a relation involving the smooth wall distance d^+_{min} and the imposed transport quantity value $\tilde{\nu}^+_w$ at the wall:

$$d^+ = d^+_{min} + \frac{\tilde{\nu}^+_w}{k} \tag{3}$$

where k is the Von Karman constant. By relating the normalized sand-grain roughness $h^+_s = \frac{h_s u_\tau}{\nu}$ to the imposed wall value $\tilde{\nu}^+_w$ using the correlations below and solving the transport equation addressed before, the shift in velocity profile is obtained.

$103595 \leq h^+_s$ $\quad \tilde{\nu}^+_w = 1.1066\ (10)^{-6}\ (h^+_s)^2 + 1.1949\ (10)^{-2}\ (h^+_s) = P$ (4)

$150.4 \leq h^+_s \leq 103595$ $\quad \tilde{\nu}^+_w = P - 6.4762\ (10)^{-12}\ (h^+_s)^4 + 1.653\ (10)^{-8}\ (h^+_s)^3$

$$- 1.279\ (10)^{-5}\ (h^+_s)^2 + 9.66\ (10)^{-4}\ (h^+_s) + 1.8067$$

$4.24 \leq h^+_s \leq 150.4$ $\quad \tilde{\nu}^+_w = P + 1.72 - 2.8 \exp\left(-\frac{h^+_s}{23.3}\right)$

$h^+_s \leq 24.4$ $\quad \tilde{\nu}^+_w = 0$

The Boeing extension is performed by defining the non-zero value of transport quantity $\tilde{\nu}$ at the wall. Here, the wall condition ($\tilde{\nu} = 0$) is replaced by a new wall condition that gives a non-zero value of transport quantity $\tilde{\nu}$ at the wall [4]:

$$\frac{\partial \tilde{\nu}}{\partial n} = \frac{\tilde{\nu}}{d} \tag{5}$$

where the wall distance is computed by ($d = d_{min} + d_0$), d_{min} being the smooth wall grid distance and d_0 is the imposed shift, which depends to the roughness value (h_s):

$$d_0 = \exp(-8.5k) h_s \approx 0.03 h_s \tag{6}$$

The other changes include modifying the f_{v1} function in the Spalart-Allmaras model, for better prediction of smaller roughness, through the new definition of χ:

$$\chi = \frac{\tilde{\nu}}{\nu} + 0.5 \frac{h_s}{d} \tag{7}$$

In the two equations k-ω model, two rough wall models are addressed: Wilcox and Knopp methods [6, 7]. In Wilcox method, a boundary condition for specific dissipation rate ω is applied to define the suitable value of ω on the wall for the imposed roughness:

$$\omega = \frac{u_\tau^2}{\nu} S_R \tag{8}$$

$$S_R = \begin{cases} \left(\frac{50}{h^+_s}\right)^2 & h^+_s < 25 \\ \left(\frac{100}{h^+_s}\right) & h^+_s \geq 25 \end{cases} \tag{9}$$

where u_τ is the friction velocity and h^+_s is the normalized sand grain roughness.

In Knopp method, the wall distance d is shifted by ($d = d_{min} + 0.03h_s$). The turbulent kinetic energy parameter at the wall k_w is computed by:

$$k_w = \varphi_{r1} k_{rough} \ , \ k_{rough} = \frac{u_\tau^2}{\beta_k^{1/2}} \ , \ \varphi_{r1} = \min\left(1, \frac{h_s^+}{90}\right) \tag{10}$$

where β is a constant equal to 0.9. The dissipation rate ω_w at the wall is computed by:

$$\omega_w = \frac{u_\tau}{\beta_k^{1/2} k d_0} \ , \ d_0 = \varphi_{r2} 0.03 h_s \tag{11}$$

$$\varphi_{r2} = \min\left[1, \left(\frac{h_s^+}{90}\right)^{\frac{2}{3}}\right] . \min\left[1, \left(\frac{h_s^+}{45}\right)^{\frac{1}{4}}\right] . \min\left[1, \left(\frac{h_s^+}{60}\right)^{\frac{1}{4}}\right] \tag{12}$$

The computed ω_w is limited compared to the ω_w value for smooth wall:

$$\omega_w = \min\left(\frac{u_\tau}{\beta_k^{\frac{1}{2}} k d_0}, \frac{60\nu}{\beta_w y^2}\right) \tag{13}$$

The Boeing extension rough wall treatment of Spalart-Allmaras is implemented in NSCODE. ONERA extension rough wall treatment of Spalart-Allmaras and k-ω model both Wilcox and Knopp rough wall models are implemented in NSMB code.

2. Grid Metrics

Here, a number of grid metrics formulations are presented [8].

Orthogonality: positive value and favorable to be close to 1.

$$\text{Orthogonality} = 1 - \left|\vec{t_i} . \vec{t_j}\right| \tag{14}$$

where $\vec{t_i}$ and $\vec{t_j}$ are the tangent unit vectors pointing out the selected point on the direction of two adjacent edges.

Skewness: positive value and favorable to be close to 0.

$$\text{Skewness} = 1 - \frac{\Delta L_{\text{Face shorter diagonal}}}{\Delta L_{\text{Face longer diagonal}}} \tag{15}$$

Stretch Ratio: positive value and favorable to be close to 1.

$$\text{I Stretch Ratio} = \frac{\Delta L_{(i_2 - i_3)}}{\Delta L_{(i_1 - i_2)}} \quad \text{and} \quad \text{J Stretch Ratio} = \frac{\Delta L_{(j_2 - j_3)}}{\Delta L_{(j_1 - j_2)}} \tag{16}$$

References

[1] Hasanzadeh K., Laurendeau E., and Paraschivoiu I., "Quasi-steady convergence of multi-step Navier-Stokes icing simulations," Journal of Aircraft, June 2013, DOI: 10.2514/1.C032197.

[2] Croce G., Candido E., Habashi W., et al., "FENSAP-ICE: Analytical Model for Spatial and Temporal Evolution of In-Flight Icing Roughness," *Journal of Aircraft*, 47(4), August 2010.

[3] Beaugendre H., Morency F., Habashi W., "FENSAP-ICE: Roughness Effects on Ice Shape Prediction," *41st Aerospace Science Meeting & Exhibit, Reno*, Nevada, Jan 2003.

[4] Aupoix B. and Spalart P. R., "Extension of the Spalart-Allmaras turbulence model to account for wall roughness," *Heat and Fluid Flow Journal*, Vol. 24, 2003, pp. 454, 46.

[5] Brahimi, P. M.T. Tran and Paraschivoiu. I. "Numerical Simulation and Thermodynamic Analysis of Ice Accretion on Aircraft Wings," Technical Report, *Bombardier Aerospace Group*, Quebec, Canada, May, 1994.

[6] Vos J.B. and Rey B., "Implementation of roughness modeling in the NSMB CFD Code," *CFS Engineering*, Technical Report, 2007.

[7] Hellsten A. and Laine S., "Extension of the k-ω-SST turbulence Model For Flows Over Rough Surface," AIAA Atmospheric Flight Mechanics Conference, New Orleans, Louisiana, 1997.

[8] Tecplot.360 2013, User's Manual, Release 1, 2013.

APPENDIX B – ADDITIONAL NSGRID AND CANICE2D-NS DATA

Content:

1. **Grid generation NSGRID results**
2. **User manual of CANICE2D-NS/NSGRID/NSCODE/NSDROP**

1. Grid generation NSGRID results

1.1. Full Multi-Grid application

Considering the transformed elliptic Poisson equations in the computational domain are as follows:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2\left(Px_{\xi} + Qx_{\eta}\right) \quad (1)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2\left(Py_{\xi} + Qy_{\eta}\right) \quad (2)$$

$$\alpha = {x_{\eta}}^2 + {y_{\eta}}^2 \quad , \quad \gamma = {x_{\xi}}^2 + {y_{\xi}}^2 \quad (3)$$

$$\beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta} \quad , \quad J = x_{\xi}y_{\eta} - y_{\xi}x_{\eta} \quad (4)$$

The transformed equations are in quasi-linear form. The equations of control functions P, Q are based on Sorenson method, as follows:

$$P(\xi,\eta) = p(\xi)e^{-a\eta} + r(\xi)e^{-c(\eta_{max}-\eta)} \quad (5)$$

$$Q(\xi,\eta) = q(\xi)e^{-b\eta} + s(\xi)e^{-d(\eta_{max}-\eta)} \quad (6)$$

where p, q, r, s are the calculated functions on the boundaries and a, b, c, d are the propagation parameters.

A range of explicit and implicit solution approaches such as Point Jacobi, Point Gauss Seidel, Point and Line SOR, Line Implicit and ADI in the context of Full Multigrid method are implemented with their stability and convergence rate examined. A number of clean and icing problems are studied to validate the elliptic grid generation: Laplace, transformed elliptic equation without/with source terms (Sorenson).

Laplace

Figure 1 shows the Multigrid (MG) and Single-grid (SG) solution convergence curves of the Laplace equations for NACA0012 clean airfoil geometry. Note that the symbols SG4, SG3, SG2, SG1 and SG0 stand for single grid solutions of the coarse (4), medium (3), fine (2), extra-fine (1)

and superfine grid (0), whereas the symbols MG2, MG3, MG4 and MG5 stand for Multigrid solutions on the extra-fine grid with 2, 3, 4 and 5 levels, respectively. In addition, the Figure contains results obtain with different smoothers: SOR, ADI, and Line implicit schemes.

Transformed elliptic equation without source terms

Figures 2 and 3 show the Single-grid solution convergence of transformed elliptic grid equations without source terms (P,Q=0) for the approaches such as Point Jacobi (PJ), Point Gauss Seidel (PGS), Point SOR (PS), Line SOR in I (LSI), Line SOR in J (LSJ), Line SOR in I and J (LSIJ), and ADI for NACA0012 clean airfoil.

Transformed elliptic equation with source terms

Figures 4-6 shows the Multi-grid and Single-grid solution of the elliptic grid equation with source terms (Sorenson control function P,Q) forced on the iced boundaries and throughout the grid domains, using the approaches: PGS, PS, LSI, LSJ, LSIJ, ADI for NACA0012 run 408 ice shape, which shows LSI providing better computation time, almost 45% faster than PS. The 2 levels V cycle Multi-grid speed up convergence with 10 relaxation steps on coarse grid is shown in Figure 6 for the PS and LSI approaches, which results LSI-FMG faster in computation time (16 seconds, almost 60% faster than PS) compared to the other analyzed methods (grid density of 257 by 129, O-mesh).

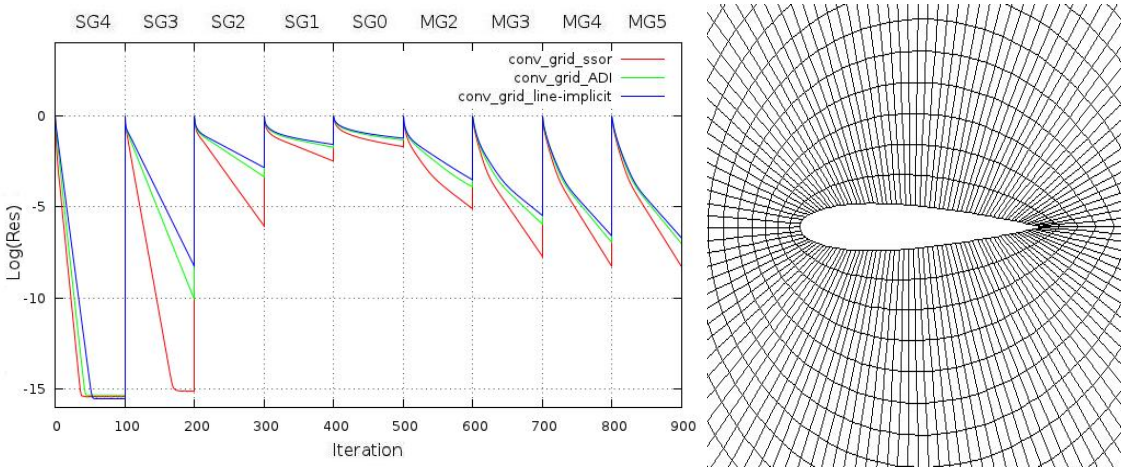

Figure 1: Laplace grid (α, γ=1 and β,P,Q=0) and FMG convergence rate (SOR, ADI, Line Implicit), NACA0012 clean (129 by 65).

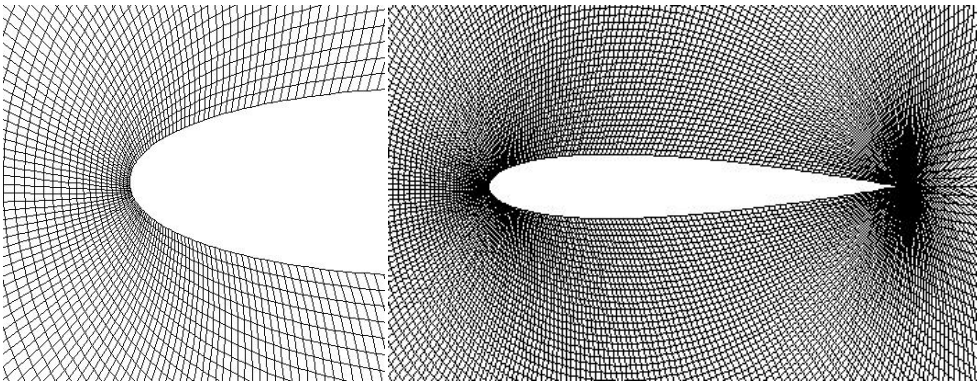

Figure 2: Ice shape elliptic smoothed grid (without control functions, P,Q=0), NACA0012 clean (257 by 129).

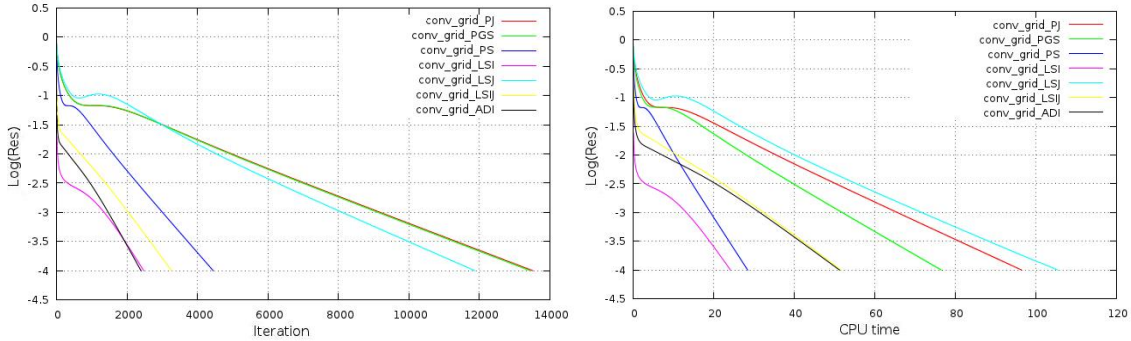

Figure 3: Convergence of grid generation (Single-Grid), (without control functions, P,Q=0), NACA0012 clean.

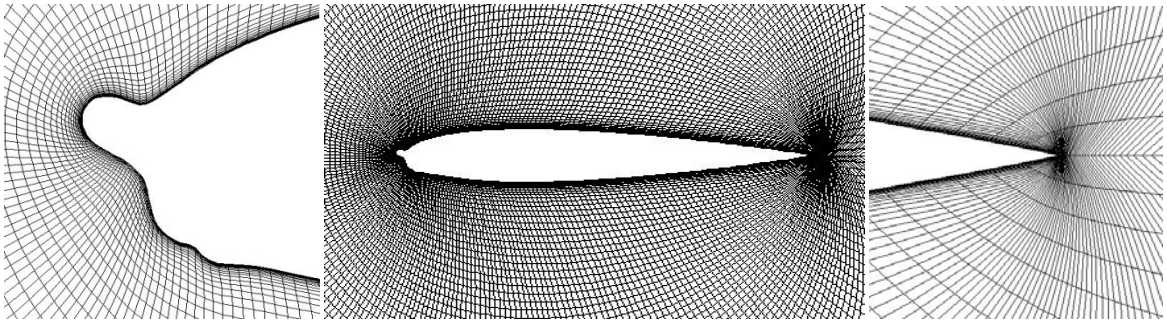

Figure 4: Ice shape elliptic smoothed grid with control functions, P,Q, NACA0012 ice run 408 (O-mesh, 257 by 129).

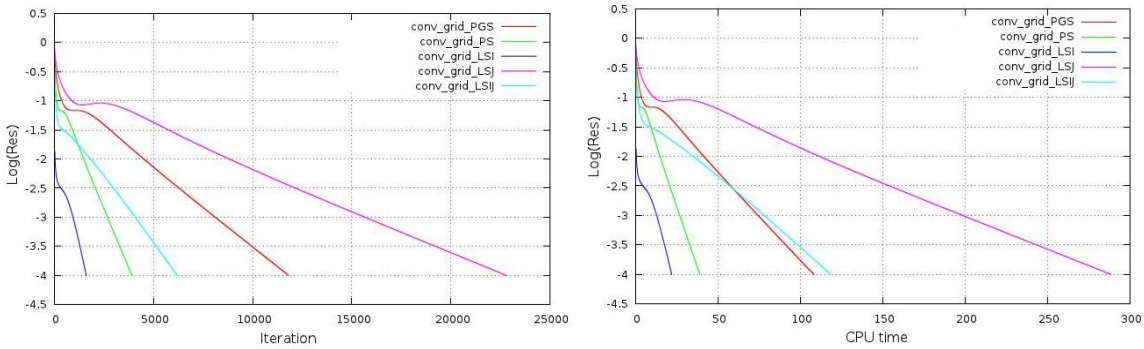

Figure 5: Convergence of grid generation with control functions, P,Q, NACA0012 ice run 408 (Single-Grid).

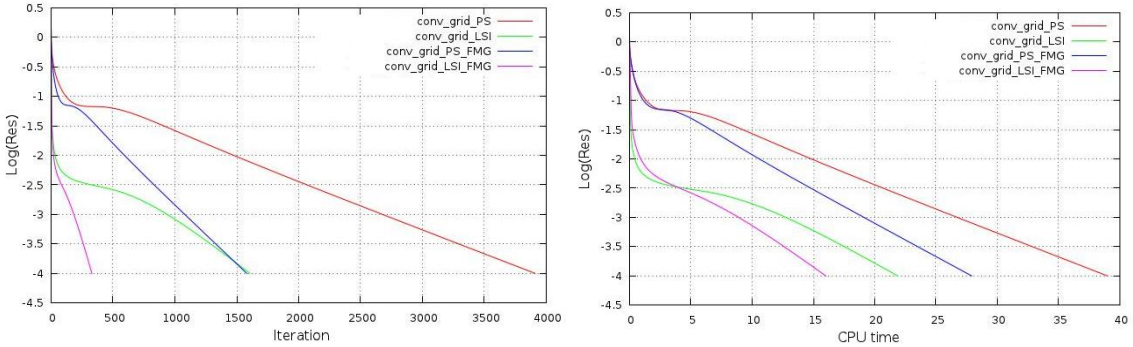

Figure 6: FMG Convergence of grid generation, with control functions, P,Q, NACA0012 ice run 408 (V cycle, 2 levels, 10 relaxation steps on coarse mesh).

1.2. Surface curvature based mesh adaptation

Experimental 2D Glaze ice 944 airfoil GLC305

The case study is the experimental ice 944 on GLC305 airfoil. A grid sensitivity analysis is studied for the developed elliptic grid generation with curvature based elliptic surface points distribution. Using NSGRID, 3 levels of grids have been generated as coarse, 257by129, medium, 513by129 and fine, 1025by129 that are shown in Figure 7. Two different surface points distributions are applied, multiple points curvature distribution based (Curv.) and single point curvature distribution based (Sin.).

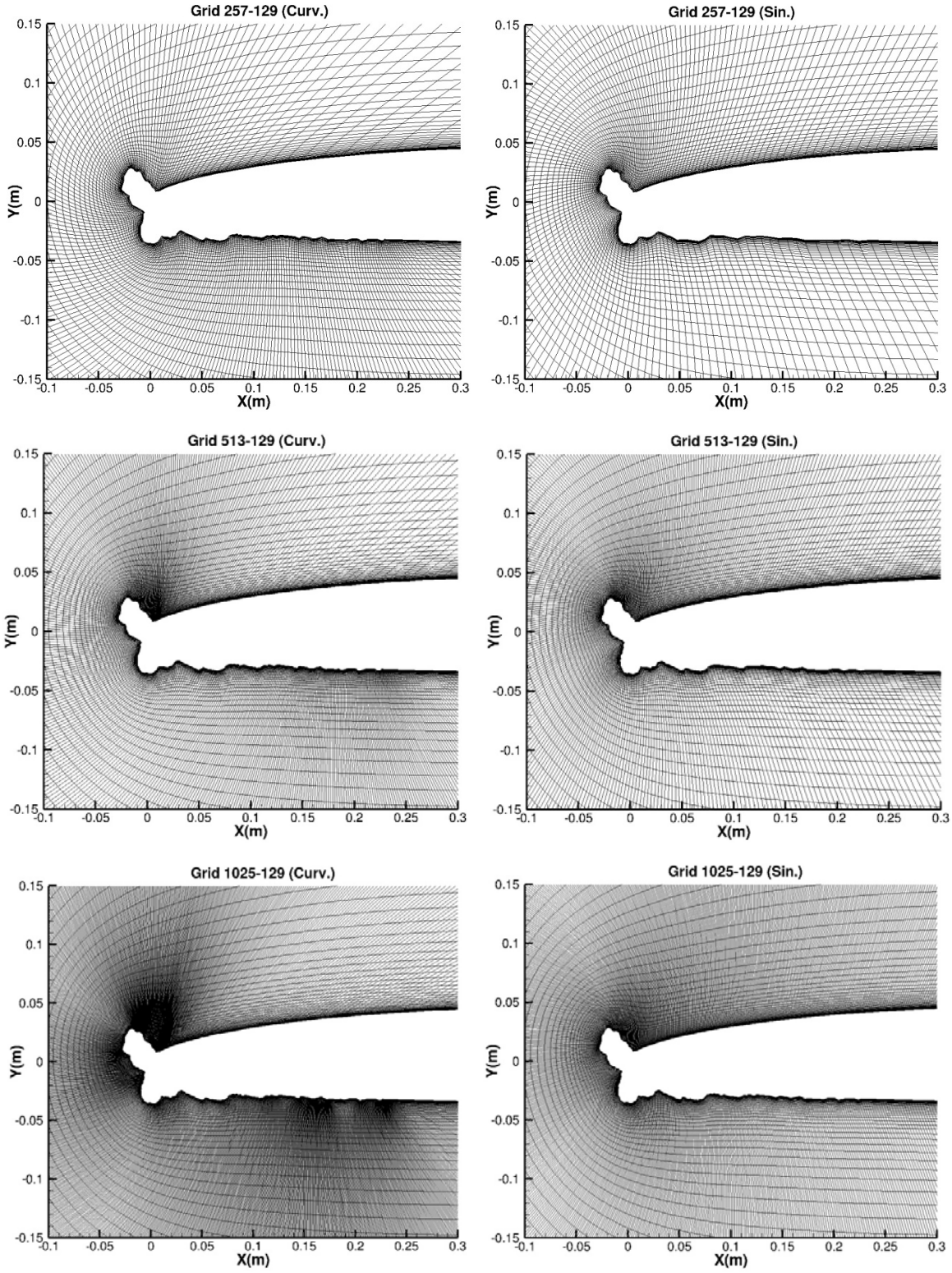

Figure 7: Elliptic grid with curvature based point distribution for case ice 944 (257-129, top; 513-129, middle; 1025-129, down) (Sorenson approach, RLS).

The curvature based point distribution (Curv.) is compared by a simple point distribution (Sin.). For the curvature based (Curv.), points with positive curvature higher than 0.1 (user input value) are chosen to apply the source terms. For the simple distribution (Sin.), the point with highest curvature is selected to apply the source terms. The ice surface curvature and the new 1D elliptic point distributions (Curv. based and Simple Sin. based, for all 3 grid levels) are shown in Figures 8 and 9, respectively. The grid generation convergence is shown in Figures 10.

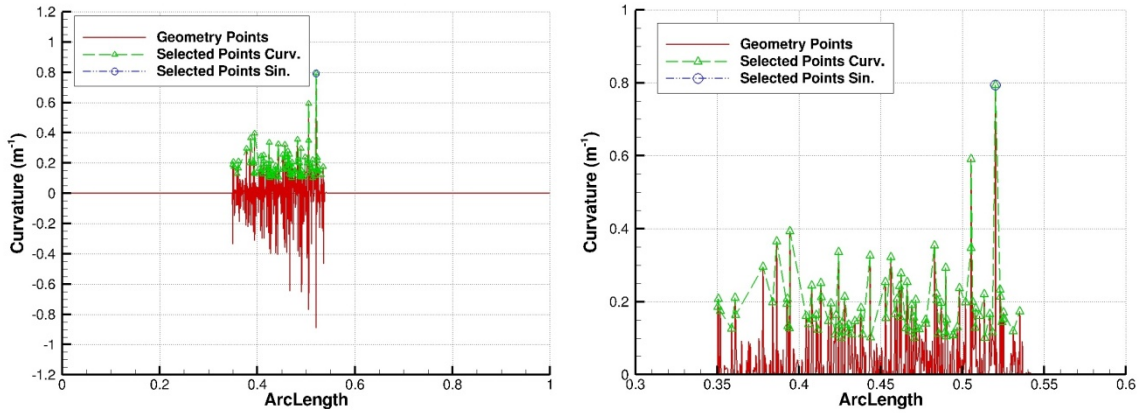

Figure 8: Ice-944 geometry point's curvature (selected points in green for Curv. based and in blue for simple Sin.), (zoom view on right).

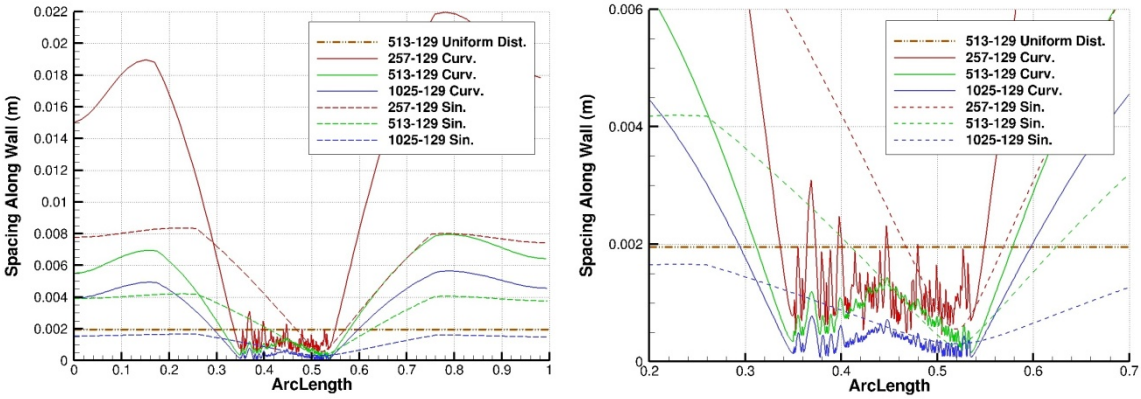

Figure 9: Curvature based distributed points spacing along the wall, (zoom view on right).

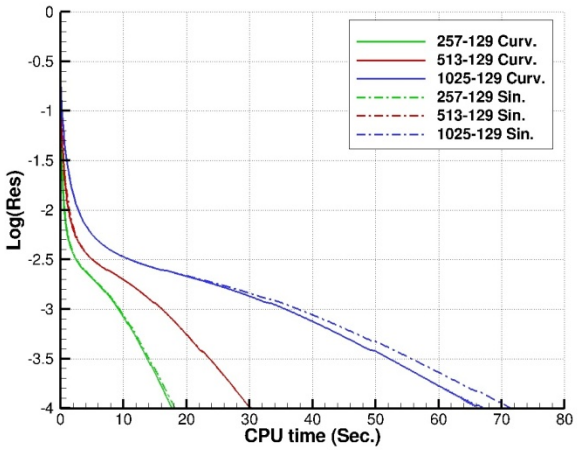

Figure 10: NSGRID solution convergence (LSOR-I, MG V-cycle) (Sorenson approach, RLS).

2. User manual of CANICE2D-NS/NSGRID/NSCODE/NSDROP

The CANICE2D icing code has been modified to interact with an automated Multi-block structured mesh generation code NSGRID, the Reynolds-Averaged Navier-Stokes (RANS) code NSCODE, and the Eulerian droplet simulation code NSDROP, developed at Polytechnique Montreal. The resulting icing code, CANICE2D-NS, is capable of automated multi-step ice accretion simulations. The computation time of the framework is successfully reduced using the FMG algorithm in the flow solver as well as the mesh generation procedure. The CANICE2D-NS multi-time steps process is based on an automated batch procedure. To run the code, one needs to have the folder (input) and the file (batch.txt). The folder (input) includes all the necessary input files and executable files of the codes (NSGRID2D, NSCODE2D, NSDROP, and CANICE2D-NS). The file (batch.txt) has two different modes to use: Lagrangian or Eulerian droplet computation. The file (batch.txt) is shown below:

```
#!/bin/sh

maxlayer=1

mkdir results
mkdir temp

######### input files ##########
cd input
```

```
cp nsgrid2d input_nsgrid2d_main input_nsgrid2d_algebraic input_nsgrid2d_smoothing input_nsgrid2d_geometry  -t ../
cp nscode2d input_nscode2d_main input_nscode2d_topo  -t ../
cp NSDROP input_NSDROP_main  -t ../
cp canice2dns input_canice2dns_main input_canice2dns_rmassl  -t ../

cd ../
#################################

for ((i=1; i<=maxlayer; i++)) ; do

mkdir layer_$i

cp nsgrid2d input_nsgrid2d_main input_nsgrid2d_algebraic input_nsgrid2d_smoothing input_nsgrid2d_geometry  -t layer_$i
cp nscode2d input_nscode2d_main input_nscode2d_topo -t layer_$i
cp NSDROP input_NSDROP_main -t layer_$i
cp canice2dns input_canice2dns_main input_canice2dns_rmassl -t layer_$i

cd layer_$i

./nsgrid2d input_nsgrid2d_main
./nscode2d input_nscode2d_main
#./NSDROP input_NSDROP_main
./canice2dns


cp output_canice2dns_geometry output_canice2dns_rmassl -t ../

mv ice.tec ice_lay_$i.tec
cp ice_lay_$i.tec -t ../results
mv beta.tec beta_lay_$i.tec
cp beta_lay_$i.tec -t ../results
mv beta_drop.tec beta_drop_lay_$i.tec
cp beta_drop_lay_$i.tec -t ../results
mv conv_grid conv_grid_lay_$i.tec
cp conv_grid_lay_$i.tec -t ../results
mv conv_flow conv_flow_lay_$i.tec
cp conv_flow_lay_$i.tec -t ../results
mv conv_drop conv_drop_lay_$i.tec
cp conv_drop_lay_$i.tec -t ../results

cd ../

mv output_canice2dns_geometry input_nsgrid2d_geometry
mv output_canice2dns_rmassl input_canice2dns_rmassl
```

```
done

mv nsgrid2d input_nsgrid2d_main input_nsgrid2d_algebraic input_nsgrid2d_smoothing input_nsgrid2d_geometry  -t temp
mv nscode2d input_nscode2d_main input_nscode2d_topo -t temp
mv NSDROP input_NSDROP_main -t temp
mv canice2dns input_canice2dns_main input_canice2dns_rmassl -t temp

date
```

This batch.txt file is for icing run with 1 layer (maxlayer=1) and Lagrangian droplet computation. To change it to multi time steps, only needs to change the variable (maxlayer) to the number of layers that is decided for icing computation. Also, it is needed to define the icing accretion time for each layer and import it to the CANICE2D-NS input file (inpu_canice2dns_main), to the section (TOTAL ACCRETION TIME (sec)). To change it to Eulerian droplet mode, only needs to activate the line (./NSDROP input_NSDROP_main), so the NSDROP code starts to run after each NSCODE2D run. Also, in CANICE2D-NS input file (inpu_canice2dns_main), it is needed to change the section (DropletTrajectory (0=Lagragian 1=Eulerian)) to 1.

In general, the batch.txt creates the folders (results, temp, and layer_1 to maxlayer). The folders (layer_1 to maxlayer) include all the data (input and computations) related to each layer (1 to maxlayer). Thus, by running the batch.txt, at first, it creates the folder layer_1 and copies the necessary initial files into it and then starts running the computation related to layer 1. When the computation of the layer 1 is completed, the batch.txt creates the folder layer_2 and this process continues until the maximum layer. At the end of the computation of all the layers, the output files (ice.tec, beta.tec, beta_drop.tec, conv_grid.tec, conv_flow.tec, conv_drop.tec) of all layers are copied in folder (results) and tagged with their layers numbers. The un-necessary files, at the end of the computation, are moved to the folder (temp).

To move from each layer to the next layer, 2 necessary output files need to be renamed and used as the input files for the next layer computations (this is done in batch.txt file). These files are (output_canice2dns_geometry) and (output_canice2dns_rmassl) which need to be renamed as (input_nsgrid2d_geometry) and (input_canice2dns_rmassl), respectively. The file (output_canice2dns_geometry) is the CANICE2D-NS generated ice shape coordinates prepared

for NSGRID2D code to produce the mesh, thus its name need to be changed to (input_nsgrid2d_geometry). The file (output_canice2dns_rmassl) is the water mass stationary on the surface at the end of the ice accretion process completed by the CANICE2D-NS. Thus, to import the amount of surface stationary water mass of the last layer to the icing computation of the next layer, the file needs to be renamed as (input_canice2dns_rmassl) which will be used by the CANICE2D-NS.

2.1. Grid generation code (NSGRID2D)

NSGRID2D input files include (input_nsgrid2d_main, input_nsgrid2d_geometry, input_nsgrid2d_algebraic, input_nsgrid2d_smoothin, and input_nsgrid2d_postprocessing). The main output files of NSGRID2D are: algebraic grid (xalg_mb), parabolic grid (xp_mb), smoothed elliptic grid in plot3d format (xs_mb) and in ascii format (xs_ascii_mb), geometry input for CANICE2D-NS (input_canice2dns_geometry), and conv_grid (convergence monitoring file). The file (input_nsgrid2d_main) is the main input defining the other input files, the initial grid, the parametric domains grid, and the maximum iteration and residual criterion for smoothing. The sample file (input_nsgrid2d_main) is as follows:

```
Airfoil
gen. algebraic grid(0-no 1-yes)
          1
input_nsgrid2d_algebraic
input_nsgrid2d_geometry
smoothing grid(0-no 1-yes)
       1
input_nsgrid2d_smoothing
read xinitial(0-n 1-y)  read st(0-n 1-y) read stp(0-n 1-y)
       1             1             1
xp_mb
st_mb
stp_mb
post-processing grid(0-no 1-yes)
```

```
        0
input_nsgrid2d_postprocessing
xs_temp
MB_iter      MB_logtoldown   MB_logtolup
 500      -4          1e3
```

The file (xp_mb) is the generated parabolic grid which is used as the initial solution. The files (st_mb, and stp_mb) are the parametric arclength domain grids generated by algebraic and parabolic physical domain grids. The total number of iteration is defined by (MB_iter), and the minimum and maximum residual criterion is defined by (MB_logtoldown) and (MB_logtolup), respectively.

The sample file (input_nsgrid2d_algebraic) is as follows:

```
Multi-Block mesh for an Airfoil
Total ijK:  it1       it2       jt1       jt2
            1         257       1         129
Number of blocks
        1
////////////////////////////////////////////////  BLOCK 1  ////////////////////////////////////////////////////
Block     i1        i2        j1        j2
  1       1      257       1         129
AlgebGrid(0-lagrange 1-hermit)   if hermit: gslop_j1  gslop_j2
      0                              1       1
AlgebST(0-lagrange 1-hermit)   if hermit: stslop_j1  stslop_j2
      0                              1       1
//////////  Lower boundary j=j1 :
Option  6,pde_dist:  solver[12-PS,16-AI]      iter        logtoldown  sweepi   omega(SSOR,ADI)
deltime(ADI)
                  12              50000       -6      0        1.5           4
```

Parabolic grid smoother: (0-no, 1-yes) wall-ds1(<=0_no, alg) aspect-ratio ttalg_i1_option(0-new 1-old)

1 0.0000008 1.067 0

Parabolic grid smoother: cf-0-0.5pi(def=1) cf-0.5pi-pi(def=1) cf-pi(def,0)

10 1 0

Parabolic grid smoother: j-start iter beta cf-rls cf-sps cf-bks

10 5 -1 1 0 0

Parabolic grid smoother: LSI-omega pe-omega qe-omega coeff-a coeff-b

0.5 0.5 0.5 0.5 0.5

Algebraic-Parabolic grid: para-extend(1-yes) j-inter option ds1 ds2

1 95 3 0.001 0.2

Algebraic-Parabolic grid: alg-papa-modif(1-yes) js-smoo je-smoo iter beta cf-rls cf-sps cf-bks

1 50 127 20 -1 1 0 0

Algebraic-Parabolic grid: LSI-omega pe-omega qe-omega coeff-a coeff-b

1 0.5 0.5 0.5 0.5

Algebraic-Parabolic grid: cf-0-0.5pi(def=1) cf-0.5pi-pi(def=1) cf-pi(def,0)

0 0 0

Read boundary points: geometry file

Coordinates: x y z

PhysicalDomainPointsDitributionCycles 2

Points ditribution 1

inumb npd mpd

1 500 3000

nst nnd mst mnd opt (opt0-5:)ds1 ids2 (opt6:)isrc isincf(0-no) ipicf ialtol <=icmin >=icmax ntol_1 ntol_n

1 500 1 3000 5 0.050 0.000050 10 0.002 4 0.0001 -100000 100000 0 0

Points ditribution final

inumb npd mpd

1 3000 257

```
 nst  nnd  mst  mnd  opt (opt0-5:)ds1  ids2  (opt6:)isrc  isincf(0-no)  ipicf  ialtol  <=icmin
>=icmax  ntol_1  ntol_n
  1  3000  1  257  6    0.0020  0.0050      10      0.0003    4    0.0001  -1000000  1000000
0      0
ParametricDomainPointsDitributionCycles  0
//////////  Upper boundary j=j2 :
Option  6,pde_dist:  solver[12-PS,16-AI]     iter       logtoldown  sweepi   omega(SSOR,ADI)
deltime(ADI)
                12              50000      -6     0       1.5          4
Read boundary points:   npd
            219
coordinates:  x   y   z
    54.2404 -5.8
    51.9846 -17.101
    48.3013 -25
    43.3022 -32.1394
    37.1394 -38.3022
    .
    .
    .
    5   50
    13.6824 49.2404
    22.101   46.9846
    30 43.3013
    37.1394 38.3022
    43.3022 32.1394
    48.3013 25
    51.9846 17.2
    54.2404 8.6824
    55 0
    54.2404 -5.8
```

PhysicalDomainPointsDitributionCycles 1

points ditribution final

inumb npd mpd

1 219 257

nst nnd mst mnd opt (opt0-5:)ds1 ids2 (opt6:)isrc isincf(0-no) ipicf ialtol <=icmin >=icmax ntol_1 ntol_n

1 219 1 257 5 0.000150 0.00250 10 0.02 2 0.0001 -100000 100000 0 0

PparametricDomainPointsDitributionCycles 0

////////// Left boundary i=i1 :

Option 6,pde_dist: solver[12-PS,16-AI] iter logtoldown sweepi omega(SSOR,ADI) deltime(ADI)

12 50000 -6 0 1.5 4

Read boundary points: npd

2

coordinates: x y z

1.000000000 0.0000000000000000000

54.2404 -5.8

PhysicalDomainoPintsDitributionCycles 1

points ditribution final

inumb npd mpd

1 2 2

nst nnd mst mnd opt (opt0-5:)ds1 ids2 (opt6:)isrc isincf(0-no) ipicf ialtol <=icmin >=icmax ntol_1 ntol_n

1 2 1 129 3 0.000000008 0.05 10 0.02 2 0.0001 -100000 100000 0 0

ParametricDomainPointsDitributionCycles 0

////////// Right boundary i=i2 :

Option 6,pde_dist: solver[12-PS,16-AI] iter logtoldown sweepi omega(SSOR,ADI) deltime(ADI)

12 50000 -6 0 1.5 4

```
Read boundary points:   npd
             2
coordinates:  x   y    z
  1.0000000000       0.0000000000000000000
    54.2404 -5.8
PhysicalDomainPointsDitributionCycles  1
points ditribution final
inumb   npd  mpd
  1     2  2
  nst   nnd   mst   mnd   opt (opt0-5:)ds1   ids2   (opt6:)isrc   isincf(0-no)   ipicf   ialtol   <=icmin
>=icmax   ntol_1  ntol_n
   1   2   1  129   3     0.000000008     0.05        10        0.02           2     0.0001  -100000  100000
0       0
ParametricDomainPointsDitributionCycles  0
```

The file (input_nsgrid2d_algebraic) defines the parameters to generate both algebraic and parabolic grids. The necessary parameters for grid generation for the application of CANICE2D-NS are presented. The file includes 4 data sections (for single block application) for different edges definitions: Lower boundary (j=j1), Upper boundary (j=j2), Left boundary (i=i1), Right boundary (i=i2). The parameters it1, it2,jt1, and jt2 are the starting and ending numbers of points defined to distribute on the edges in i and j. Parameters i1, i2, j1, j2 are also the number of points on the edges which need to be defined as (i1=1, i2=257, j1=1 and j2=129). The parabolic grid input parameters are in the sections (parabolic grid smoother, and algebraic-parabolic grid). The important parameters are: (j-inter) is the j line intersection of parabolic and algebraic; (js-smoo) and (je-smoo) are the starting and the ending j lines to smooth the grid in j, respectively. The input file (input_nsgrid2d_algebraic) is set by default to generate both algebraic and parabolic-algebraic grid.

The section (PhysicalDomainPointsDitributionCycles) defines the properties of the point distribution on the edges. The first distribution is for curvature computations and the second distribution is for grid generation. The important parameters are npd and mpd which are the number of point before and after the distributions. Parameters nst, nnd, mst, and mnd are the

starting and ending numbers of points before and after the distribution. The parameter opt defines the option of point distribution: 0 for the same distribution as initial data; 1 for the lower side concentration, 2 for the upper side concentration, 3 for both sides concentration, 4 for central concentration, 5 for uniform concentration, and 6 for PDE curvature based distribution. The parameters (ds1 and ds2) are used to control the spacing close to the starting and ending points for the options (1,2,3,4). To use option 6 (PDE curvature based distribution), the important parameters are: isrc to define the source term (isrc=10, by default, is the sin(x) source term); isincf is the source terms factor which defines the ratio of points concentration and propagation; ipicf value is 2 for the curvature based point distribution (and 4 for simple sin(x) distribution to concentrate the points only in leading edge and trailing edge); the parameter (<=icmin) defines the minimum curvature value for selecting the geometry points to apply the source terms (the geometry points with the higher curvature value are selected to concentrate the points around); the parameter (>=icmax) defines the maximum curvature value for selecting the geometry points to apply the source terms (the geometry points with the lower curvature value are selected to concentrate the points around). The section (Read boundary points: npd) and (coordinates: x y z) define the number of initial points and their coordinates.

The initial geometry points for the ice surface are defined in file (input_nsgrid2d_geometry). The file (input_nsgrid2d_geometry) includes: the name of the airfoil (first line); the airfoil chord value (second line); number of points (third line); and the geometry points coordinates. The sample file (input_nsgrid2d_geometry) is as follows:

```
 NLF0414
 0.9
  216
  1.002000     -2.750795E-02
  1.000000     -2.7513999E-02
 0.9986255     -2.7411999E-02
 0.9966345     -2.7280301E-02
 0.9938821     -2.7125200E-02
 0.9902904     -2.6961500E-02
 0.9866976     -2.6824800E-02
```

.

.

.

0.9690340 -2.2205999E-02

0.9761550 -2.3456000E-02

0.9823700 -2.4491999E-02

0.9876600 -2.5333000E-02

0.9912001 -2.5880700E-02

0.9939670 -2.6298899E-02

0.9961652 -2.6621999E-02

0.9983661 -2.6926700E-02

1.000000 -2.7122000E-02

1.002000 -2.750795E-02

The file (input_nsgrid2d_smoothing) includes all the parameters related to the definition of the grid smoother source terms and the multi-grid solver. The sample file (input_nsgrid2d_smoothing) is as follows:

Multi-Block Mesh for an airfoil

TOTAL ijK: it1 it2 jt1 jt2

1 257 1 129

Number of Blocks EDGsmooth_ValueForAllBlocks(-1,no)

1 -1

// BLOCK 1 //

Block i1b i2b j1b j2b BLKsmooth(0-no 1-singleBLKsmooth 2-MultiBLKsmooth)

1 1 257 1 129 1

FreeEdgesNumb 2

Block Edge is ie js je EDGsmooth(0-no 1-yes)

1 1 1 257 1 1 0

1 3 257 1 129 129 0

ConnectivityEdgesNumb 2 EDGsmooth_ValueForAllEdges(-1,no): -1

Block Edge is ie js je EDGsmooth(0-no 1-mod1 2-mod2)

1 2 257 257 1 129 0

1 4 1 1 1 129 0

1 4 1 1 129 1 0

1 2 257 257 129 1 0

////////// smoothing solver options:

mg mode(0-lin, 1-nonlin) mg type[0-V(1,0), 1-V(1,1), 2-w(1,1)] coarsest_relax relax_mg_pre relax_mg_post

1 1 25 1 1

FMG(FMG: 0-j, 1-ij) Interpol-opt(0-lin1 1-lin2) P,Q interpolation (0-solve on each level 1-restrict from level_0 2-only on level_0, 0 on other levels)

1 0 1

residual smoothing[0-no, 1-yes] source_smooth: (0-no 1-yes) sc_solver[12-PS] sc_sweepi sc_omega(SSOR,ADI) sc_deltime(ADI)

0 0 12 1 1.5 10000

solverapp(0-allitc 1-eachitc) sweepi(-1,0,+1) sweepj(-1,0,+1) solver[10,20-PJ,11,21-PGS,12,22-PS,13,23-LSI,14,24-LSJ,15,25-LSIJ,16,26-AI,17,27-AJ,18,28-AIJ]

1 +1 +1 13 0

logtoldown logtolup boundsmooth(0-no 1-Omeah 2-Cmesh)

-5 1e3 0

P,Q selection (0-no 1-sigma 2-ro_si 3-RLS 4,5-SPS0 6,7-SPS 8,9-FEQ 10-RLS-SPS 12-ORTHO-D)

10

Othogonality-Numann(0-no 1-yes) relax-f

0 1

(if P,Q selection = 0) LaplacEq(0-no 1-yes) alpha beta gamma (active if >=0)

0 1 0 1

(if P,Q selection = 1) sigma_p(in kesi) sigma_q(in eta)

0.001 0.1

(if P,Q selection = 2) options[1-ro,si_const., 2-ro,si_form1(min), 3-ro,si_form1(max), 4-ro,si_form2] (opt=1)ro_constant si_constant

1 0 0.05

(if P,Q selection = 3) source(0-Eta_min 1-Eta_max 2-both)

0

spacing_Eta_min(0-input 1-alggrid 2-fromfile),value spacing_Eta_max(0-input 1-alggrid 2-fromfile),value (if P,Q = 3)

1 0.000001 0 0.01

angle-Eta_min(0-input 1-fromfile),value angle-Eta_max(0-input 1-fromfile),value (if P,Q = 3)

0 90 0 90

Coeff_a(Eta_min) Coeff_b(Eta_min) Coeff_c(Eta_max) Coeff_d(Eta_max) (if P,Q = 3)

0.2 0.3 0.5 0.5

omega_pe(Eta_min) omega_qe(Eta_min) omega_re(Eta_max) omega_se(Eta_max) (under-relax[0_1]) (if P,Q = 3)

0.02 0.002 0.2 0.02

(if P,Q selection = 7 SPS-Orthogonal) method(10-FDM, 11-FDM2, 20-FVM, 21-FVM2) solver(0-SOR 1-LSOR-I 2-LSOR-J 3-LSOR-IJ)

11 0

iteration SweepI SweepJ omega logtol

2000 1 1 1.5 -8

(if P,Q selection = 8,9 FEQ) WA(Area) WL(length) WO(orthogonality) (with WA+WL+WO=1, WA,W,WO>=0)

0.0 0.8 0.2

(if P,Q selection = 10 RLS_SPS) cf_rls: P,Q cf_sps: P,Q (if sps: 0-s,c 1-s,c,o)

1 1 0.5 0.5 0

(if P,Q selection = 10 RLS_SPS) cf_sps_orth: P,Q a_sps_orth,b_sps_orth i_1,coef_sin_1 i_n,coef_sin_n ji_1_n j_n,coef_sin_jn

0.001 0.001 0 0 0 0 0 0 0 0 0

(if P,Q selection = 12 ORTHO-D) dlt-ortho

0.5

closewall grid gen: cwg_active(0-no 1-mode1 2-mode2 3-mode3) mode1: cwg_subnum cwg_ratio cwg_wallsp_tol

0 4 2.5 0.0000000001

closewall grid gen(mode 2): hmtimes linebreaksnumerby ratio

9 2 0.5

closewall grid gen(mode 3): pointnumber option j1minsp j2minsp

129 3 0.000001 0.1

itc 1

level iter mglevel solver(if solvapp=1) sweepi sweepj omega(SSOR,ADI) deltime(ADI)

0 1 2 26 1 1 1 10000

The parameters it1, it2,jt1, and jt2 are the starting and ending total numbers of points defined to distribute on the edges in i and j. Parameters i1b, i2b, j1b, j2b are the number of points (for each block) on the edges which need to be defined as (i1b=1, i2b=257, j1b=1, j2b=129). The section (FreeEdgesNumb) defines the edges for the application of the smoother RLS source terms. The section (ConnectivityEdgesNumb) defines the connectivity between the domain edges. The file is set to use the blended approach source terms (the parameter (P,Q selection) is equal to 10) and to use 2 levels of multi-grid. The blended approach uses source terms RLS, SPS, and ortho. The input files for RLS source terms are defined in section (if P,Q selection = 3). The main important parameters are: Coeff_a(Eta_min) and Coeff_b(Eta_min) that define the propagation of the RLS source term into the domain (by default, they are set to 0.2 and 0.3, respectively); omega_pe(Eta_min) and omega_qe(Eta_min) are the under-relaxation factors for the source terms (by default, they are set to 0.02 and 0.002, respectively).

The important parameters of the blended approach are in the section (if P,Q selection = 10). They include: the weight factors (cf_rls: P,Q) for RLS source term (by default, they are set to 1, 1); the weight factors (cf_sps: P,Q) for SPS source term (by default, they are set to 0.5, 0.5); the weight factors (cf_orth: P,Q) for parabolic source term (by default, they are set to 0.001, 0.001). At the end, the parameter (solver(if solvapp=1)) defines the type of solver to use (value 26 defines ADI in I). The parameters (omega) and (deltime(ADI)) are the relaxation and delta-time factors in (Point and Line SOR) and (Point-Jacobi and ADI), respectively. The options for the

solver selections are: 20-PJ; 11-PGS; 12-PS; 13-LSI; 14-LSJ; 15-LSIJ; 26-AI; 27-AJ; 28-AIJ. The preferable solvers are LSI (13) with relaxation of 1 and ADI-I (26) with delta-time of 10^{10}.

2.2. Flow solver code (NSCODE2D)

The main input file for NSCODE2D run is (input_nscode2d_main) which reads the other input files: the grid topology file (input_nscode2d_topo) and the grid file (xs_mb). The output files of NSCODE2D are: level_0 (flow solution), level (flow velocity and density solution for CANICE2D-NS), cp (pressure coefficient, skin friction solution on the surface and density close to the wall for CANICE2D-NS), conv_flow (monitoring file), and the file (input_nscode2d_main.air) for NSDROP code. The sample of the file (input_nscode2d_main)is as follows:

Topology file: input_nscode2d_topo
Mesh file: xs_mb
mach alpha cl dcl reynolds
0.2876 0.0 no 0.001 6.95e6
Alpha ramping (1-yes/0-no initial_alpha MGcycles)
0 0. 100.
flow(0 steady 1 unsteady) dt tmax
0 .2e0 1000.
xref yref cmac
0.125 0.0 0.9
dissip vis2 vis4 (1-first order dissipation 2-JST dissipation 3-MATD)
2 1/2 1/32
V_n V_l V_nc V_lc switching coefficients (for MATD only). c indices indicate coefficients for coarse meshes.
0.3 0.3 0.4 0.4
Runge-Kutta scheme (1-Explicit 2-Point-Jacobi)
1
a
1

```
turbulent method (0-no 1-baldwin_lomax 2-spalart_allmaras 3-k_omega) Transition (0-no 1-yes)
2                    0
a b
1 1
viscosity ratio    FSTI
100.                .5
roughness method (0-no 1-boeing 2-ONERA)  hs
1                                    0.0001
residual smoothing (0-no 1-yes)
1
vortex correction (0-no 1-yes)
1
read restart (no or filename)
no
write restart (1-yes or 0-no)
0
coarsening (1 - i&j coarsening   0 - j coarsening)
0
itc
1
level  iter    mglevel rk  cfl  logtol  a  b c
0      600      4     5   6.5    -4       2  1
seg_coup  Freeze_mode   Freeze_numb Trans_relax F_corel
0         3             500           0.1         0
```

The important parameters are: mach, alpha (angle of attack), Reynolds number, cmac (chord value), dissip (choice of dissipation, preferable JST), vis2 (second order dissipation), vis4 (forth order dissipation), turbulent method (preferable spalart_allmaras), roughness method (preferable Boing), hs (roughness value), J coarsening (for multi-grid application, preferable 0 (coarsening in j only) for icing problem), iter (number of iterations), mglevel (number of multi-grid levels), cfl (preferable 7.5), and logtol (minimum residual criterion).

2.3. Eulerian droplet solver code (NSDROP)

The main input file for NSDROP is (input_NSDROP_main) which reads the files: the grid file in ascii format (xs_ascii_mb), and the flow solution file (input_nscode2d_main.air) that made by NSCODE2D. The main output files of NSDROP are: the droplet beta and water solution for CANICE2D-NS (beta_water_drop), the droplet convergence monitor (conv_drop). The sample of the file (input_NSDROP_main) is as follows:

```
AIRFOIL
meshtopology 0                [0-O_mesh_1-C_mesh]
 257x 129  itl 1 itu 257          [imax_x_jmax_itl_itu]
xs_ascii_mb
airflow filename
input_nscode2d_main.air
mach alpha reynolds  altitude  tinf
0.2876   0.0   6.95e6   10000.0    257.59
xref yref cmac
0.25   0.   0.9
MVD         LWC  (droplet parameters)
20e-6  0.33
dissip vis2
0       1.0
residual smoothing (0-no 1-yes)
0
runge kutta update scheme (0-explicit 1-point-jacobi)
1
Add source term  (0-no 1-yes)
1
Limiter  (Vl      Vn     Vs  psi)
      0.1 1.   0.1     1.
itc
```

1

level iter mglevel rk cfl LOGTOL

0 500 1 5 0.8 -5

The main parameters are: mach, alpha (angle of attack), Reynolds number, altitude, tinf (temperature), cmac (chord value), MVD (Mean Equivolumetric Diameter), LWC (Liquid Water Content), Limiter (set as defaults), iter (number of iterations), cfl (set as default), LOGTOL (minimum residual criteria).

2.4. RANS based icing code (CANICE2D-NS)

The input files of CANICE2D-NS are: (input_canice2dns_main) that includes the icing parameters, (input_canice2dns_rmassl) that includes the stationary water mass on the surface from the icing last layer, (input_canice2dns_geometry) that is generated by NSGRID2D and defines the geometry data for CANICE2D-NS, (level) that is generated by NSCODE2D and defines the flow velocity and density, (cp) that is generated by NSCODE2D and defines the flow properties on the surface, (beta_water_drop) that is generated by NSDROP and defines the Eulerian beta and water solution. The main output files of CANICE2D-NS are: (beta.tec) is the Lagrangian beta solution, (ice.tec) is the computed ice shape solution, (output) is the main computed parameters data, (I-SXC-XC-CF-CP-UEIM-ST-HC-IMCFD and I-SXC-XC-RKS-RHOOAA-CPAIR-IMCFD) are the heat transfer computation parameters data. The sample of the main input file (input_canice2dns_main) is as follows:

********** FLIGHT CONDITIONS **********

DropletTrajectory (0=Lagragien 1=Eulerien)

1

MODE OF CANICE2D-NS (1=CFDFLOW OR 2=CFDFLOWTHERMO)

2

CFCOEFF RHOmodif CF-UEsmooth(0=No, 1=Yes) niter I+ I-

1 0 1 10 5 5

ANGLE OF ATTACK(deg.) OR Cl (ILIFT = 0 OR 1)

```
   0   0.0
CHORD (0=m , 1=inch)
   0   0.9
FREESTREAM VELOCITY (0=m/s , 1=ft/s , 2=Knots(KEAS) )
   0      92.54
ALTITUDE ISA (0=m, 1=ft)
   0      0.0
USE OF AMBIENT TEMPERATURE(0) OR TOTAL TEMPERATURE(1)
   0
AMBIENT TEMPERATURE (0=Celsius, 1=Fahrenheit)
   0    -15.56
LIQUID WATER CONTENT (g/m**3)
   0.33
MEAN EQUIVOLUMETRIC DIAMETER (m)
   20.0E-6
ROUGHNESS HEIGHT (m) GIVEN BY USER OR CALCULATED (IKS = 0 OR 1)
   0.0001       0
TOTAL ACCRETION TIME (sec)
   1224
NUMBER OF ICE LAYERS
   1
_____________________________________________
********** TRAJECTORIES **********
_____________________________________________
Y INCREMENT FOR TRAJECTORIES AND ACCURACY REQUIRED(Runge-Kutta 5)
   0.0005      1.0D-04
Droplet Release point (x times of chord)
   5
_____________________________________________
********** OPTIONS **********
_____________________________________________
```

TRANSITION POINT 1st BODY[0=AUTO, 1=MANUAL, INTRADOS (S/C) , EXTRADOS (S/C), 2=IMPINGEMENT LIMITS]

0 0.06 0.06

USE SMOOTHING FOR ICE SHAPE CALCULATION (YES=1,NO=0)

0

USE OF THERMODYNAMIC ANALYSIS (YES=1,NO=0)

1

OUTPUT FILES (.OUT=0, .TEC=1)

1

PRINT TRAJECTORIES (YES=1,NO=0)

1

PRINT HEAT AND MASS TERMS (YES=1,NO=0)

1

USE OF FINITE DIFFERENCE CALCULATION FOR THE FIRST BODY (YES=1,NO=0)

0

USE OF HIGHLIGHT FOR CURVILINEAR DISTANCE IN BETA (YES=1, NO=0)

0

USE OF METER OR INCH FOR COORDINATES IN PANEL.OUT (METER = 0, INCH = 1)

0

********** ANTI ICING **********

USE OF ANTI-ICING PROCEDURE(YES=T,NO=F)

F

CONVERGENCE CRITERIA FOR RESIDUE

1.D-03

MAXIMUM NUMBER OF ITERATION FOR ANTI-ICING PROCEDURE

250

RELAXATION FACTOR FOR SURFACE TEMPERATURE AND HEATFLUX

5.D-01 5.D-02

STARTING POINT OF THE ANTI-ICING DEVICE (NODE NUMBER)

150
END POINT OF THE ANTI-ICING DEVICE ON THE INTRADOS (PANEL)
100
END POINT OF THE ANTI-ICING DEVICE ON THE EXTRADOS (PANEL)
200
NAME OF THE FILE FOR THE CONVECTION COEFFICIENT ON THE ANTI-ICING SIDE
q53.dat
WALL CONDUCTIVITY (W/(m K))
1.00
WALL THICKNESS(m)
0.2D-02
USE OF INITIAL TEMPERATURE FILE (YES=1, NO=0,NAME)
0 dataanti.in

The main parameters are: (DropletTrajectory) that needs to set to 0 for the case of Lagragian droplet computation and 1 for the case of Eulerian droplet calculation, (CF-UEsmooth) which defines the choice of skin friction smoothing (1 for activation), (niter) is the number of iteration for smoothing, (I+ and I-) are the number of selected panels before and after the stagnation points for smoothing the skin friction, and the other defined parameters as (ANGLE OF ATTACK), (CHORD), (FREESTREAM VELOCITY), (AMBIENT TEMPERATURE), (LIQUID WATER CONTENT), (MEAN EQUIVOLUMETRIC DIAMETER), (ROUGHNESS HEIGHT), (TOTAL ACCRETION TIME), (Y INCREMENT FOR TRAJECTORIES).